

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

ISTENENLER ve ALGORITMA

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 OBEB'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 gelinmişse elimizdeki sayı listenin en başına eklenir. Eğer ilerleme sonucunda listenin başka bir yerine gelindi ise, bu defa sayımız gelinen sıranın sağ tarafına eklenir. Listeyi gezerken, elimizdeki sayı ile listedeki hiçbir sayının OBEB'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 gelinen 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, gelinen 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 yapılı 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. Diğeri 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();
    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 gezebilmek 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.

```
iosya Düzenle Ara Görünüm Kodlama Diller Ayarlar Araçlar Makrolar Çalıştır Eklentiler Pencereler ?

🔒 🔒 陆 🐿 😘 😘 🚵 🏅 🏗 🏗 🕽 🚅 🛍 🏂 💜 🤏 🗓 📆 📜 🖫 🖫 🕦 💹 💯 🚳 🎾 🗎 🗈 🕦
🔜 Node hpp 🖾 🛗 CircularDoublyLinkedList.hpp 🖾 🛗 ProgramController.hpp 🖾 🛗 CircularDoublyLinkedList.cpp 🖸 🛗 Program
                     ### CircularDoublyLinkedList.cpp
### 8description Dairesel cift yonlu bagil liste clusturarak of
#### 8course | Dairesel cift yonlu bagil liste clusturarak of
#### 8course | Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored Colored 
                                                                                                                                                                                                                                                                Try the new cross-platform PowerShell https://aka.ms/pscore6
                                                                                                                                                                                                                                                                PS C:\Users\Acer> cd .\Desktop\
PS C:\Users\Acer\Desktop> cd .\Vize_Projesi_B181210397\
PS C:\Users\Acer\Desktop\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?yA8AAJKKKOU\SRb
                 #include<iostream>
#include "CircularDoublyLinkedList.hpp"
                 using namespace std;
                          CircularDoublyLinkedList::CircularDoublyLinkedList()
                                                                                                                                                                                                                                                                   Sifre: A2_?yA0AWAJK4KKOUfU\s9S`Rbbx
                  // OBEB bulma islemi
int CircularDoublyLinkedList::obeb(int sayi1, int sayi2)
                                                                                                                                                                                                                                                                   Sifre: yAAAKSR
                                   int buyukSayi;
                                   if(sayi1 > sayi2)
  buyukSayi = sayi1;
                                   else
buyuksayi = sayi2;
                                 for(buyukSayi; buyukSayi > 0; buyukSayi--) (
   if( (sayil % buyukSayi == 0) 66 (sayi2 % buyukSayi ==
        break;
                                    )
                                        return buyukSayi;
                       // Gelen item'ın kurulan algoritma ile CDLL'e eklenmesi
yoid CircularDoublyLinkedList::insert(int item){
   int mod = 0;
```

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

```
losya Düzenle Ara Gorunüm Kodlama Diller Ayarlar Araçlar Makrolar Çaliştir Eklentiler Pencereler 2
3 🔐 🔛 🐿 😘 😘 🚵 🔏 🎋 🐿 🖎 🗷 🛍 🏖 🕊 🤏 🔯 🛂 🚍 🖫 🎵 🌃 🐔 🐿 💌 🗷 🕮
                                                                                                🔞 🤮 CircularDoublyLinkedList.cpp 🗵 🔛 ProgramController.cpp 🔀 🖺 Progra
                  search = " ";
currPos = 0;
k = 0;
prevPos = 0;
sayilar = new CircularDoublyLinkedList();
                                                                                                                                     ifre: }Kww4wKe,||;8256H8S89.OBE<2H0VhTDMF7>\S6FYPQ;FCXOz74O]7rEDt;Jloo>AWtCQ;rraHs/7^+29rDf^a]A@ru
`CqA1Qb}u9U74sM\8|]B?q0@x8>K9YZou~PVTqCCBcFH{q|xDCJWoa*SF7k}MCdF6{dxP7uhzDzuxBs}6FN+94T0FxEgh45]AW
Q8s/r^Xao+Swj_tR3Mv^QVn_d~Q+e3tirf`Vko+j_f[il_aifcg3hij33g
                                                                                                                                  Sifre: DELV<D77Y]X9dPF1=a^DBggH\h=-~~~~>*N<6I]w/KbOj>N;3=|OXY-m^>[N;W7Jq.>/:jqhcYV:CM2V1,gjY16E]<a
K^e]Fi4VNQ{B5j.+P@zxzZ>V61LaaJND?3vb{E?;gxPO0M8<LJ*QUjS381}_?`e<BAmn@Rai4oz_:IQ]3Z.Kvu:56*D{WfJ0HIG
q]sLQAIj\XqGbm|H3ZY4]._.RcaXe//eZs}fxp{u1fq9ng|9kyru9psqqrs
                        {
    spacePos = oneLine.find(search, currPos);
    if(spacePos >= 0) {
        currPos = spacePos;
        stringstream turnStream(oneLine.substr(prevPos turnStream >> arr[k];
        currPos++;
    prevPos = currPos;
}
                                                                                                                                   Sifre: CDOSPdSAPSJiGKT[SAQ`}}<4+QLj;.iZXa75d.5d>W^X]Caksjo]`~~~b[Vma5PP7RfC*qKkjy`yyZ;X<vbvC/=6wV<
pf=x<-B;DCA3][=@p+H1K1/CC{3]-?P*BA.B=9DGDJA.[29<Ee,-DM5[m],Y_2M2i{1K33a4a+c3Y||8b->t1Nh`?|@Q?f1}.:>
p^N<+Sx36zjfT\\oM,8.,|6,98Wh99:qc{2^p8qon8p{uho7j1zrwxxu9st
                                                                                                                                   Sifre: Xp@e^qqN{|}Y/zykYnldFUOfqLc*cmSm`+1S44gw|qXkYcaoaoax`Linwg5ka{a|ttiL7wkros1=Xg032kk];+h+o3/i
31;MU4[yU|,]}1}\^p]HeRmtwkob{kz^{LyPEX],01_6eM7cGgqGGjp7kQ-,4I7vQ]?1L2300ef{,9fH\}TUhNYYZAo/SEi456M
cx71L2oi*8Z?FbbK9UU-ZHBV~04KLFQLEWQU<LLK8>4=MFEJAABBC>A>?~?
                 ) while (spacePos >= 0);
                  stringstream turnStream (oneLine.substr(prevPos, o turnStream >> arr[k];
                                                                                                                                                  <zzf70+C<OinnoSaIH]C\2.}0HFLJZ9VhOwRCXCTU|`2**.MK=^OEq_YZZJo.d\1Z*3-~qN|;msOV50},z[=Paq-<.\
                                                                                                                                   lW,\wOOh[4Ez8VIhEzm~0xN87TA87V*Bpe?586Kw*+Jmy=fdC5*II3BLBB~;_t`X}KdeZaX+RKfdk90BGnD.k\h
|Mj40uYFLrhh2C_zJr^R.5UsR|u06kXX{v\ov~/\0mza6ijwehppplqlono
                                                                                                                                   Sifre: ffFusGacazoQ-15}hpVwBCo|v3wnwsp^<.mDS_@ENmyt6:29fiq-7Gxs?7d/Gl-g4G[Nqrh?=ced5Lbft/FB//vapcRP
yGKF@}6zzG`)-tVmOT;yd}S`R{VK;38J\JoK~+`IQnmQK388698aBgO@7W=oeM8|Pr-E@*1B+=kQ|dG:ut_I.`5]LPpZdZxr,RK
y?]fx2<D6jcwC0Cn^VX|W+Najbsc,Qui+RFKXPEV*ZV^V_NPO+XT_XVW[YY
                delete sayilar;
      void ProgramController::calistir()
                                                                                                                                 Sifre: 6p:Kj\R`N3VR_DhNTGK1PP0yn0YJXmo.JJYSg`h^GJ8aOof`L:k[lskcmj}}cV6uqwg}Ah;8W9jVd>^vUlt[:xoooojm
+QYvyqTv:y`bNZZItXu<pCW\vdP1-KnH,*G/iok^M*QO,J2Fr,T<B7ZX\AUTFYAX*B_IP7M9YRTTKG@P/78-TLIYO/CMc]H\F,6
a{d_MG.~G*]]zBU[?c4C41,d4=m<6;mApb1767fjj|t|ns=q{{<<<;xw<zz
                  // Sayilar.txt dosyasini okuma modi
ifstream readFile("Sayilar2.txt");
string newLine = "";
                                                                                                                                  Sifre: B0;cCDXAggC>KI00GGHi0=ZfZUzz?zO*LM8<TL+|||B>l=/,oL;06,1BIX6=fJU5jj/A?+15>,UX`Z;dN[Ral<xL06B1
~>.eoCMP5-SUZ7M.JSRgw/..wcI4?}ll1INx4k_Dw*akP^eN@1NCd>LyVaoo_Xa>qFyCy7*g{t/mYX*z7IdN]E]6qG-;Lht3;ez
YKZqrG+{7oQdH\8PU,eod10o+v]6e6qp8se::~kq82qvitk|lumzwtwrwuu
                        Dosyanin acilip acilmadigini kontrol edi
( readFile.is_open() ) {
  while ( getline(readFile, newLine) ) {
    split(newLine);
    cout << endl << endl;
                                                                                                                                     ress any key to continue
```

YAPILANLAR, ZORLANILAN KISIMLAR ve SONUÇ

Projeyi yaparken 4 dosyaya bölmek istedim. Main (*Program.cpp*) dosyası ve 3 adet header dosyası. Listemizin bulunduğu (*Node.hpp*) dosyasında sadece listenin struct yapısını oluşturdum. Liste ile ilgili işlemlerin yapıldığı dosyada (*CircularDoublyLinkedList.hpp*), listemize ekleme ve yazdırma işlemlerini tanımladım. Programın çalışması için çeşitli kontrolleri yaptığım dosyada (*ProgramController.hpp*) ise, string'e ait **split** ve **calistir** fonksiyonlarını icra ettim. Böylelikle daha düzenli bir kod yapısı hedefledim ve bu dosyaları main bloğumuzun bulunduğu dosyaya dahil edip işlemleri gerçekleştirdim.

C++ gibi makine diline yakın dillerde string işlemleri yapmak, diğer programlama dillerine kıyasla biraz daha zor olabiliyor. Bir metin içerisinde arama, kesme, değiştirme gibi işlemler tek komutlarla olamayabiliyor. Bu yüzden bu proje içerisinde, metin dosyasından satırdaki sayıları aldıktan sonra işlenmesi kolay olmadı. Bunun için internette bolca araştırma yapmam gerekti. Yaptığım araştırma sonucunda bir, iki farklı yol buldum. Ve anlaması en kolay yolu seçerek projeye dahil ettim.

Sonuç olarak; dosyadaki satır sayısı ve her satırdaki sayı adedi değişken sayıda olan iki farklı dosyayı (Sayilar.txt, Sayilar2.txt) okuyup, **iki yönlü dairesel bağıl liste** ile en büyük obeb algoritması kullanmamla, satırda bulunan sayıların ASCII karşılıklarına göre her satırın bir şifreyi temsil etmesini sağlayarak bir C++ konsol uygulaması geliştirmiş oldum. Projeyi verilen algoritmaya ve istenilen özelliklere göre tamamladım. Eksik bir kısım kalmadı.

KAYNAKÇALAR

- [I] https://medium.com/programmercave/doubly-circular-linked-list-c-implementation-850165897321
- [2] https://www.yusufsezer.com.tr/cpp-dosya-islemleri/
- [3] https://www.softwaretestinghelp.com/stringstream-class-in-cpp/

Not: İşbu rapor şahsıma ait olup, yararlandığım kaynaklar yukarıdaki gibidir. Kimseyle paylaşmadığımı belirtmek isterim.