



**SAKARYA**  
**ÜNİVERSİTESİ**

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİ YAPILARI 1. ÖDEV**

**HAZIRLAYAN: Ecem AMANVERMEZ**

**G181210022 / 1A**

**Öğretim Görevlisi: Dr.Öğr.Üyesi MUHAMMED FATİH ADAK**

Ödevde istenilen faktöriyel hesaplaması normal veri tiplerine sığmadığı için, sayılar ArrayList veri tipinde saklanılıyor, sayılar Listede ters yönden yerleştiriliyor, yani sayının ilk basamağı listenin birinci indeksine yazılıyor, bu sayı büyüdükçe son indekse ekleme yapılıyor, aksi taktirde sürekli sayıları kaydırmak gerekli ki bu büyük performans kaybına sebep olur.

ArrayList.hpp dosyasında ArrayList sınıfı için tanımlanmış + operatorü

```
friend ArrayList* operator+(ArrayList& leftt, ArrayList& right);
```

ArrayList.cpp dosyası bu fonksiyonun gövdesini içeriyor.

İki ArayList'in toplamını hesaplamak amaçlı yazılan kısım:

```
ArrayList* operator+(ArrayList& left, ArrayList& right)
{
    int overflow = 0;
    int i = 0;
    ArrayList* result = new ArrayList();

    while (i < left.size() && i < right.size()) {
        result->add((left.elementAt(i) + right.elementAt(i) + overflow) % 10);
        overflow = (left.elementAt(i) + right.elementAt(i)) > 9 ? 1 : 0;
        i++;
    }
    while (i < left.size()) {
        result->add((left.elementAt(i) + overflow) % 10);
        overflow = (left.elementAt(i) + overflow) > 9 ? 1 : 0;
        i++;
    }
    while (i < right.size()) {
        result->add((right.elementAt(i) + overflow) % 10);
        overflow = (right.elementAt(i) + overflow) > 9 ? 1 : 0;
        i++;
    }
    if (overflow == 1) result->add(1);

    return result;
}
```

Test.cpp dosyasında ilk olarak kullanıcıdan bir sayı alınıyor INT SAYI değişkenine atanıyor,

```
int sayi;
cin >> sayi;
```

Sonra For döngüsü 1’den başlayarak girilen sayıya kadar faktöriyel çarpanına atanıyor.

Döngüde iki ArrayList tanımlanıyor “tmp newList”

tmp listesine faktöriyel çarpanının her basamağı çarpma işlemi yapılarak atanıyor, ve sonrasında tmp ve newList’in toplamı newListin içine atanıyor, faktöriyel çarpanın sonuncu basamağının çarpma işlemi bittiğinde newList -> list değişkenine atanıyor.

```
delete list;  
list = newList;
```

faktöriyel çarpanı girilen sayıya ulaştığında döngü biter ve faktöriyel hesaplaması tamamlanmış olur,

Sonra sonucu içeren list değişkeni Sonuc.txt dosyasına yazılır.

Yazma işleminde kullanılan << operatörü sayının düzgün şekilde yazılması için listeyi son indeksinden birinci indeksine yazdırır.

```
ostream& operator<<(ostream& screen, ArrayList& right)  
{  
    for (int i = right.lenght - 1; i >= 0; i--) {  
        screen << right.items[i];  
    }  
    return screen;  
}
```

Her new anahtar kelimesi ile tanımlanan liste işlem bittiğinde delete ile iade edilir.