**Algoritma ve Programlama Projesi**

**Mehmet Mert Fidan 19360859018**

**Levent Burak Çil 20360859012**

Oluşturduğumuz projenin amacı girilen input'un karakterlerini, oluşturduğumuz İngilizce ve Almanca dizilerindeki karakterler ile karşılaştırıp uyumluluk yüzdesine göre ekrana "Girdiğiniz İnput İngilizce'dir" ya da "Girdiğiniz İnput Almanca'dır" çıktısı vermesi.

Projemizin kullanım alanları oyunlar, çeviri siteleri, alışveriş siteleri vs. olabilir. Bu program sayesinde kullanıcılar kullandıkları siteleri/programları daha etkin ve rahat kullanabilirler.Diyelim ki kullanıcı bir textin dilini bilmiyor ve öğrenmesi gerek. Programımaza texti girip dilini öğrenebilir. Şimdilik sadece Almanca ve İngilizce karşılaştırma yapan programımız, dizi çeşitliliği arttırılıp bir çok dil için kullanılabilir hale getirilebilir.

Projemizi oluştururken Dev C++ ile CodeBlocks IDE'lerini kullandık. İlk olarak programın algoritmasını kurarak işe başladık. Kurduğumuz algoritmanın amacı oluşturduğumuz Almanca ve İnglizce dizilerinde bulunan elemanları tek tek girilen textin içinde aramaktı ve inputun hangi dizi ile benzerlik göstediğini bulmaktı.

İlk olarak matematiksel işlemler için gereken kütüphaneyi , Boolean için gereken makroları içeren kütüphaneyi ve içinde bir çok makro ve fonsiyonu bulunduran stdlib.h kütüphanesini dahil ettik. Sonra dizilerimizi oluşturduk.

İşlemlerimize klavyeden alınan texti harfler haricindeki karakterlerden arındırmakla başladık.Bunun için bir filtreleme fonksiyonu oluşturduk. Bu fonksiyonda tanımladığımız 3 adet değiken ile for döngüsü oluşturduk. Bu döngüler ile a'dan z'ye olan bütün harfler haricindeki karakterleri filtrelemiş olduk.

İkinci ve üçüncü oluşturduğumuz fonksiyonlarda dizilerde bulunan elemanların text içindeki sıklığına baktık. Bunun için iç içe for döngüleri kullandık. Metindeki bütün harfler sırayla taranıyor. Eğer metindeki bir harf ile bigram ve ya trigram listelerindeki seslerden birinin ilk harfi aynı ise bir sonraki for döngüsüne giriyor ve o harfin yanındaki harf ile karşılaştırılan bigram ve ya trigramın 2. harfiyle karşılaştırılıyor. Eğer ki metin bir trigram ile karşılaştırılıyorsa döngü aynı sekilde bir daha çalışıyor. Ardından metinde bir bigram ve ya trigram bulunduysa onun bulunğu listedeki değeri 1 artırılıyor. Bütün metnin taranması bitince metinde kaç tane bigram bulunduysa bigram listesindeki bütün sayılar o sayıya bölünüyor, böylece metindeki bütün bigram ve trigram sayıları bulunmuş oluyor.

Bir sonraki fonksiyonun içinde (calculate\_distances), bulmuş olduğumuz frekans değerlerini İngilizce ve Almanca dillerinin frekans değerleriyle karşılaştırıyor ve girilen metnin hangi dile daha benzer olduğunu hesaplıyoruz. Bunun için öklit uzaklık bağıntısını kullandık. Kısaca anlatmak gerekirse sırasıyla İnglizce ve Almanca dilleri için önce o dilin frekans değerlerini bulduğumuz frekans değerlerinden çıkarıyoruz. Ardından bütün bulduğumuz değerlerin karesini alıp topluyoruz. Son olarak topladığımız değerlerin karekökünü alıyoruz ve metnin o dile olan yakınlığını hesaplıyoruz ve program bir sonraki fonksiyona geçiyor (detect\_lang).

(detect\_lang)Adı üstüne bu fonksiyonun yaptığı tek şey metnin her iki dile olan uzaklıklarını karşılaştırmak ve yakın olan dilin İngilizce mi yoksa Almanca mı olduğunu ekrana yazdırmak(Sayısal değer olarak küçük olanı yazdırıyor).

Projeyı oluşturma sürecinde filtreleme fonksiyonunu oluşturma aşamasında karşılaştığımız hatanın çözümü için Github üzerinden araştırma yaptık aradığımız bilgiye ulaşamayınca Stackoverflow'dan öğrenebildik. Ayrıca derste öğrendiğimiz İkili arama'da (binary search) işimize oldukça yaradı. Diğer takıldığımız yerlerde ders notlarıma dönerek yapmaya çalıştık yine de yapamadıysak Github'dan örnek kodlara bakarak projemizi tamamladık.