



T.C.

AREL ÜNİVERSİTESİ

2023-2024 BAHAR YARIYILI

ALGORİTMALAR VE PROGRAMLAMA DERSİ

DÖNEM PROJESİ

**KONU: RASTGELE ÖĞRENCİ VE SINAV
NOTLARI ÜRETEN PROGRAM**

HAZIRLAYAN

KADİR CAN GEYİK

Konu: Rastgele öğrenci ve sınav notları üreten program:

Bu projede kullanıcı 0 ile 2500 arası öğrenci sayısını seçecek ve daha sonra kullanıcıdan D-Y şeklinde 10 soruluk bir cevap anahtarı seçmesi istenecektir. Kullanıcının seçtiği öğrenci sayısına göre öğrenciler adı, soyadı, yaşı ve notu şeklinde oluşturulacaktır. Daha sonra her öğrenciye rastgele soru cevapları atanacak ve daha sonra kullanıcının seçtiği cevap anahtarı ile karşılaştırılıp öğrencinin notu hesaplanacaktır. Son olarak öğrenciler adı, soyadı, yaşı, notu şeklinde kullanıcının isteğine göre sıralanıp ekrana yazdırılacaktır.

YÖNTEM

- **Kütüphaneler ve Yapı Tanımlamaları:** Kütüphaneler (vector,algorithm,iostream,string,fstream) tanımlanır. Öğrenci adında bir yapı tanımlanır ve bu yapı adı, soyadı, yaşı, cevaplar ve notu gibi öğrenci bilgilerini içerir.
- **Dosya İşlemleri:** DosyadanIsimOkuyalım fonksiyonu, belirtilen dosyadan isimleri okumak için kullanılmıştır.
- **Cevap Anahtarı Alımı:** CevapAnahtari fonksiyonu, kullanıcıdan 10 harf içeren bir string cevap anahtarı alır.
- **Öğrenci Oluşturma:** OğrencileriOlustur fonksiyonu, kullanıcının belirttiği sayıda öğrenciyi rastgele ad ve soyad dosyalarına göre oluşturur. Aynı öğrencinin tekrar etmemesi için bir kontrol eklenmiştir.
- **Soruların Cevaplarını Oluşturma:** SorularinCevaplariniOlustur fonksiyonu, her öğrenci için 10 adet olacak şekilde doğru/yanlış sorularına rastgele cevaplar oluşturur.
- **Puan Hesaplaması:** PuanlariHesapla fonksiyonu, öğrencilere ait cevapları kullanıcının girdiği cevap anahtarıyla karşılaştırıp hesaplar.
- **Öğrenci Sıralaması:** Oğrencilerisirala fonksiyonu, kullanıcının tercihi göre öğrencileri sıralar.
- **Ana Program:** Kullanıcıdan öğrenci sayısı alınır ve belirtilen aralıkta olup olmadığı kontrol edilir. Ad.txt ve soyad.txt dosyalarına göre isimler dosyalardan okunur ve vektörlere kaydedilir. Öğrenciler oluşturulur, soruların cevapları belirlenir ve puanlar hesaplanır. Kullanıcıdan sıralama tercihi alınır ve öğrenciler sıralanarak adı, soyadı, yaşı, notu şeklinde ekrana yazdırılır.

ANALİZ

- **Veri Yapıları:** Öğrenci adlı bir yapı kullanılarak öğrenci bilgileri (Adi, Soyadi, Yasi, Cevaplar, Not) bir araya getirilmiştir. Vector veri yapısı, dinamik boyutlu dizileri kullanmak için yani öğrenci ve isim listelerini tutmak için kullanılmıştır.
- **Fonksiyonlar:** Dosya İşlemleri: DosyadanIsimOkuyalım: Bir dosyadan isimleri okumak için kullanılır. CevapAnahtari: Kullanıcıdan cevap anahtarı alır. OğrencileriOlustur: Belirtilen sayıda rastgele öğrenci oluşturur. SorularinCevaplariniOlustur: Her öğrenci için belirtilen sayıda rastgele cevap oluşturur. PuanlariHesapla: Her öğrencinin puanını hesaplar. Oğrencilerisirala: Kullanıcı tercihinə göre öğrencileri sıralar.
- **Ana Program (main):** Kullanıcıdan öğrenci sayısı alınır ve belirtilen aralıkta olup olmadığı if sorgusu ile kontrol edilir. Ad ve soyad dosyalarından isimler dosyalardan okunur ve vektörlere kaydedilir. Öğrenciler oluşturulur soruların cevapları belirlenir ve puanlar hesaplanır. Kullanıcıdan sıralama tercihi alındıktan sonra öğrenciler sıralanarak ekrana sırasıyla yazdırılır.
- **Güvenlik ve Kullanıcı Kontrolü:** Program kullanıcıdan geçerli bir öğrenci sayısı girişi bekler ve girilen değeri kontrol eder. Dosya işlemleri sırasında dosyanın açılıp açılmadığı kontrol edilir. Aynı öğrencinin tekrar eklenmemesi için kontrol mekanizmaları eklenmiştir.
- **Rastgele Veri Oluşturma:** Öğrenci isimleri ve soyadları, ad.txt ve soyad.txt dosyalarından okunarak rastgele seçilir. Öğrenci yaşları ve cevapları program tarafından rastgele oluşturulur.
- **Kod Okunabilirliği:** Kod açıklayıcı isimlendirmeler ve yorumlar içerir. Fonksiyonlar modüler bir şekilde tasarlanmıştır her biri belirli bir görevi yerine getirmektedir. Kod içindeki belirli yerlerde hata kontrolleri ve kullanıcıya bilgi verme işlemleri yapılmıştır.

KOD

```
#include <iostream> //kullanıcıdan veri alıp göndermek için
#include <vector> //dinamik boyutlu diziler için
#include <algorithm> //karşılaştırmalar için gerekli
#include <string> //string dizileri için
#include <fstream> //dosya giriş/çıkış işlemleri için
using namespace std; //cout cin gibi girdi çıktı işlemleri için

//Ogrenci adında bir yapı oluşturuyoruz struct
Ogrenci { string Adi; //Öğrencinin adı dosyadan
okunacak string Soyadi; // Öğrencinin soyad
dosyadan okunacak int Yasi =0; // 18-60 arası
random oluşturulacak
string Cevaplar; //Random şekilde 10 adet D-Y harf DYDYDYDYDY şeklinde
oluşturulacak string Sonuclar; //Öğrencinin verdiği cevaplara göre
karşılaştırma sonrası her soru için doğrumu yanlış mı yapmış? int Not = 0;
//Öğrencinin notu karşılaştırma sonrası nota eklenecek };

//Dosyadan isim okumak için kullanılan fonksiyon... void DosyadanIsimOkuyalım(const
string& dosyaIsmi, vector<string>& yazilacakListe) { ifstream
dosyam(dosyaIsmi.c_str()); if (!dosyam.is_open()) {
cout << dosyaIsmi << " dosyasi acilmiyor lutfen tekrar deneyiniz ! " << endl;
}
for (string satirim; getline(dosyam, satirim); )
{ yazilacakListe.push_back(satirim); } }
//Bu fonksiyonda kullanıcıdan string bir cevap anahtarı girilmesi istenecek ve okunacak
void CevapAnahtari(string& cevapAnahtarim)
{ cout << "Lutfen 10 harften olusan (D ve Y harfleri) bir cevap
anahtari giriniz.(Örnek format DYDYDYDYDY):\n " << endl; cin >>
cevapAnahtarim;
} void OgrencileriOlustur(vector<Ogrenci>& Ogrenciler, int ogrenciSayisi,
const vector<string>& adlar, const vector<string>& soyadlar) {

// for döngüsü ile kullanıcının girdiği sayıya göre öğrencileri
oluşturuyoruz for (int i = 0; i < ogrenciSayisi; ) { Ogrenci
ogrenci;
ogrenci.Adi = adlar[rand() % adlar.size()];
ogrenci.Soyadi = soyadlar[rand() % soyadlar.size()];
ogrenci.Yasi = rand() % 43 + 18;

// ayni ogrenciden olmamasi icin bir durum tanımlıyoruz.
bool ayniOgrenciVar = false;

// for döngüsüyle öğrencileri geziyoruz aynı öğrenci var ise öğrenciyi
eklemeyecek yok ise vektöre push edecek, ekleyecek. for (const Ogrenci&
kayitliOgrenci : Ogrenciler) { if (kayitliOgrenci.Adi == ogrenci.Adi &&
kayitliOgrenci.Soyadi == ogrenci.Soyadi) { ayniOgrenciVar =
true;
// True ise öğrenciyi ekleme
// ogrenci sayisini bir arttır çünkü öğrenci eklenmedi ve for dongümüzden 1
ogrenci boşa çıktı.
```

```

        break;
    }
}

if (!ayniOgrenciVar)
{
    // false ise öğrenciyi vektöre push et...
    Ogrenciler.push_back(ogrenci);
    // ogrenci eklendiyse i'yi arttır
    ++i;
}

// Ekledikten sonra ekrana i'yi ve eklenen öğrencinin bilgilerini
yazdırarak programın ilerleyip ilerlemediğini kontrol edebiliriz //cout <<
"Eklenen Ogrenci Sayisi: " << i << endl;
//cout << "Eklenen Ogrenci Bilgisi: " << ogrenci.Adi << " " << ogrenci.Soyadi << "
" << ogrenci.Yasi << endl;
}
}

//Bu fonksiyonda her öğrenci için soru sayısına göre random cevap
oluşturuyoruz. void SorularinCevaplariniOlustur(vector<Ogrenci>& Ogrenciler,
int soruSayisi) {
    //for döngüsü ile öğrencileri gezip öğrenci yapılarımıza cevapları
    ekliyoruz for (Ogrenci& ogrenci : Ogrenciler) { ogrenci.Not =
    0; for (int i = 0; i < soruSayisi; ++i) { char cevap =
    rand() % 2 == 0 ? 'D' : 'Y'; ogrenci.Cevaplar += cevap;
    }
    } }

//Bu fonksiyonda öğrencilerin verdiği cevaplar ile kullanıcıdan aldığımız cevap anahtarını
karşılaştırıyoruz
void PuanlariHesapla(vector<Ogrenci>& Ogrenciler, const string& cevapAnahtari) {
//öğrencilerin cevap anahtarlarını geziyoruz öğrencilerin cevapları ile kullanıcının
cevaplarını karşılaştırıyoruz
    for (Ogrenci& ogrenci : Ogrenciler) { for
    (int i = 0; i < cevapAnahtari.size(); i++) {
        //eğer öğrencinin cevabı kullanıcının verdiği cevaba eşit ise,
        //öğrencinin notuna 10 puan ekle Sonuc dogru ise Sonuclara 'D' ekle
        if (ogrenci.Cevaplar[i] == cevapAnahtari[i]) { ogrenci.Not +=
        10; ogrenci.Sonuclar += 'D';
        }
        //değil ise sonuclara 'Y' ekle i değişkenini 1 arttırıp sonraki
        cevaba geç else { ogrenci.Sonuclar +=
        'Y';
        }
    }
    }
}

//Bu fonksiyonda öğrencileri Ad,soyad,yaş veya notuna göre sıralıyoruz.
void Ogrencilerisirala(vector<Ogrenci>& Ogrenciler, char sirala) {
    //switch ile kullanıcıdan aldığımız char ifadeye göre herhangi bir case'i
    çalıştırıyoruz.
    //Kullanıcı geçer veya yanlış bir sıralama tanımlar ise default ile ada
    göre sıralıyoruz. switch (sirala) { //ada göre sıralama

```

```

        case 'A': case 'a':          sort(Ogrenciler.begin(), Ogrenciler.end(), [](const auto&
k1, const auto& k2) { return k1.Adi < k2.Adi; });          break;
        //soyada göre sıralama      case 'S': case 's':          sort(Ogrenciler.begin(),
Ogrenciler.end(), [](const auto& k1, const auto& k2) { return k1.Soyadi < k2.Soyadi; });
break;
        //yaşa göre sıralama        case 'Y': case 'y':          sort(Ogrenciler.begin(),
Ogrenciler.end(), [](const auto& k1, const auto& k2)
{ return k1.Yasi < k2.Yasi; });          break;
        //nota göre sıralama        case 'N': case 'n':          sort(Ogrenciler.begin(),
Ogrenciler.end(), [](const auto& k1, const auto& k2) { return k1.Not > k2.Not; });
break;
        //default olarak ada göre sıralama
default:
        sort(Ogrenciler.begin(), Ogrenciler.end(), [](const auto& k1, const auto& k2) {
return k1.Adi < k2.Adi; });          break;
    } } int
main() {

        int ogrencisayisi = 0; //ogrencisayisi değişkenini öğrenci sayısını kullanıcıdan
almak için tanımlıyoruz      int sorular = 10; //kaç adet soru olmasını istiyoruz ?
int dogrusayisi = 0;
    //burada do while döngüsü kullanıyoruz.
    //Kullanıcının 0 ile 2500 arası haricinde bir sayı girdiğinde
    //hata bilgisi vermek ve tekrar girmesini istediğimiz için kullanıyoruz.
do {          cout << "Lutfen ogrenci sayisi giriniz ! 0 ile 2500
arasi:\n";
        //öğrencisayisini oku
cin >> ogrencisayisi;
        //eğer öğrenci sayısı 0 ile 2500 arasında değilse kullanıcıya bilgi ver
if (ogrencisayisi <= 0 || ogrencisayisi > 2500) {          cout << "Gecersiz
ogrenci sayisi! Lutfen tekrar giriniz." << endl;
}
    } while (ogrencisayisi <= 0 || ogrencisayisi > 2500); //0 ile 2500 arası değilse
öğrencisayisi tekrar do'nun içerisine gir.

        //Ad ve soyad dosyalarının adlarını tanımlıyoruz
const string adDosyasi = "ad.txt";      const string
soyadDosyasi = "soyad.txt";

        //Ad ve soyad vektorlerini oluşturuyoruz...
vector<string> adlar;      vector<string>
soyadlar;

        // Dosyalardan random isim ve soyisimler almak için dosyaları oku ve adlar soyadlar
vektörlerimize tanımla
        DosyadanIsimOkuyalim(adDosyasi, adlar);
        DosyadanIsimOkuyalim(soyadDosyasi, soyadlar);

        //Ogrencileri oluşturuyoruz...

```

```

vector<Ogrenci> ogrenciler;
OgrencileriOlustur(ogrenciler, ogrencisayisi, adlar, soyadlar);
//Soru cevaplarını oluşturunuz...
SorularinCevaplariniOlustur(ogrenciler, sorular);

//Cevap anahtarını okuyoruz...
string cevapAnahtari;
CevapAnahtari(cevapAnahtari);

//Puanları hesaplıyoruz..
PuanlariHesapla(ogrenciler, cevapAnahtari);
//Öğrenciler arasında Sıralama yapıyoruz... char siralama;//kullanıcıdan
siralamaya yapma şeklini girmesini ve bunu okumak için tanımlıyoruz. (Daha sonra
Ogrencilerisirala fonksiyonumuza göndereceğiz.)
cout << "Ogrencileri nasıl sıralamak istersiniz ?\nAda Gore = A\nSoyada Gore = S\nYasa
Gore = Y\nNota Gore = N:\n" << endl;
//kullanıcıdan bir sıralama şekli girmesini istiyoruz
cin >> siralama;

//Kullanıcı tercihi göre ogrencileri sıralıyoruz ogrencileri ve kullanıcıdan
aldığımız siralamaya değişkenini fonksiyonumuza gönderiyoruz...
Ogrencilerisirala(ogrenciler, siralama);

//Son olarak sonuclari sıraladıktan sonra ekrana uygun biçimde yazdırıyoruz...
for (int i = 0; i < ogrenciler.size(); ++i) {

    const Ogrenci& ogrenci = ogrenciler[i];
    cout << "[" << i + 1 <<
".Ogrenci] - [Adi-Soyadi: " << ogrenci.Adi + ogrenci.Soyadi << "]" - [Yasi: " <<
ogrenci.Yasi << "]" - [Puani: " << ogrenci.Not << "]" - [Sonuc: " << ogrenci.Sonuclar +
"]" << endl;
}
}

```

```
Microsoft Visual Studio Hata Ayıklama Konsolu

Lutfen ogrenci sayisi giriniz ! 0 ile 2500 arasi:
10
Lutfen 10 harften olusan (D ve Y harfleri) bir cevap anahtari giriniz.(İrnek format DYDYDYDYDY):
DYDYDYDYDY
Ogrencileri nasil siralamak istersiniz ?
Ada Gore = A
Soyada Gore = S
Yasa Gore = Y
Nota Gore = N:
A
1.Ogrenci] - [Adi-Soyadi: EmineDogan] - [Yasi: 24] - [Puani: 50] - [Sonuc: YDBYDDYDY]
2.Ogrenci] - [Adi-Soyadi: FatihYuksel] - [Yasi: 42] - [Puani: 20] - [Sonuc: YDYDYDYDY]
3.Ogrenci] - [Adi-Soyadi: FurkanAktas] - [Yasi: 30] - [Puani: 60] - [Sonuc: DDDYDYDYDY]
4.Ogrenci] - [Adi-Soyadi: HuseyinAydin] - [Yasi: 47] - [Puani: 20] - [Sonuc: DYYYDYDYDY]
5.Ogrenci] - [Adi-Soyadi: MerveKoc] - [Yasi: 31] - [Puani: 10] - [Sonuc: YYYYYDYDY]
6.Ogrenci] - [Adi-Soyadi: MerveAcar] - [Yasi: 49] - [Puani: 40] - [Sonuc: YDDYDYDYDY]
7.Ogrenci] - [Adi-Soyadi: MeryemGuler] - [Yasi: 46] - [Puani: 70] - [Sonuc: DDDDDYDYDY]
8.Ogrenci] - [Adi-Soyadi: SerkanSahin] - [Yasi: 41] - [Puani: 60] - [Sonuc: DDDYDYDYDY]
9.Ogrenci] - [Adi-Soyadi: YasinKose] - [Yasi: 47] - [Puani: 40] - [Sonuc: YDDYDYDYDY]
10.Ogrenci] - [Adi-Soyadi: YilmazYildirim] - [Yasi: 19] - [Puani: 40] - [Sonuc: YYDYDYDYDY]

C:\Users\kadir\Desktop\AlgoritmaProje\AlgoritmaDönemProjesi\x64\Debug\AlgoritmaDönemProjesi.exe (27772 işlemi), 0 koduyla
çıkış yaptı.
Hata ayıklama durduğunda konsolu otomatik olarak kapatmak için Araçlar->Seçenekler->Hata Ayıklama->Hata ayıklama durduğu
nda konsolu otomatik olarak kapat seçeneğini etkinleştirin
Bu pencereyi kapatmak için herhangi bir tuşa basın...
```