

BURSA TEKNİK ÜNİVERSİTESİ / BİLGİSAYAR MÜHENDİSLİĞİ
ALGORİTMALAR VE PROGRAMLAMA DERSİ GÜZ DÖNEMİ PROJE ÖDEVİ

18/12/2022

HAZIRLAYANLAR	
ENES BABEKOĞLU	20360859113
SANEM COŞKUN	21360805922

PROJE ÖN BİLGİLENDİRMESİ

Projemizde bizden C programlama dilini kullanarak kullanıcıdan bazı girdiler alarak (öğrenci sayısı, soru sayısı, boş bırakma ihtimali ve doğru cevap verme ihtimali) bu girdiler neticesinde projede belirtilen koşullara uygun olacak şekilde rastgele bir sınav oluşturup bu sınavın puanlandırmasını ve harflendirmesini yapan algoritmalar kurmamız istenmektedir.

ADIM 1 – Kütüphanelerin Tanımlanması

Rastgele sayı üretimi için `<time.h>` ve formüllerde yer alan karekök işlemi ile üs alma işlemini gerçekleştirmek için gereken `<math.h>` kütüphanelerini projemize dahil ediyoruz.

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4  #include <math.h>
```

ADIM 2 – Global Değişkenlerin Tanımlanması

Bazı global değişkenlerin veri tiplerine göre tanımlamasını yapıyoruz. Burada `duzeyler_min[9]` ve `duzeyler_max[9]` isimli dizileri T-Skor tablosunda harflerin indis numaralarını kullanarak sınıf düzeyine göre o harfe denk gelen puan aralığını bu 2 diziye kaydedeceğimiz için tanımladık.

```
5
6  int N, S;
7  double B, D, HBN[100], ortalama, std, T_skoru[100], duzeyler_min[9], duzeyler_max[9];
8
```

ADIM 3 – Main Fonksiyonu

Bu fonksiyon programımızın ana fonksiyonudur. Çok fazla karışık olmaması için içerisinde farklı fonksiyonları dahil ederek kullandık. Tek boyutlu `char cevap_anahtari[100]` isimli 100 karaktere kadar veri tutabilen bir karakter dizisi ve 100 öğrenciye kadar 100 cevap tutabilen iki boyutlu `char ogrenci_cevaplari[100][100]` isimli bir karakter dizisi oluşturduk.

Tek tek bize gereken hazırlamış olduğumuz tüm fonksiyonları main fonksiyonumuza dahil ettik. Tüm sistemi fonksiyonlar yardımıyla kurguladık. Kullanıcıdan girdileri `sınav_olustur()` fonksiyonunu kullanarak alarak karmaşıklığı minimuma indirdik.

```

448
449 void main(){
450
451     char cevap_anahtari[100], ogrenci_cevaplari[100][100];
452
453     sinav_olustur(); // Sınav Girdilerini Alıyoruz
454     cevap_anahtari_uret(cevap_anahtari, S); // Cevap Anahtarı Üretiyoruz
455     cevap_anahtari_yazdir(cevap_anahtari, S); // Cevap Anahtarını Konsola Bastırıyoruz
456     sinavi_uygula(ogrenci_cevaplari, cevap_anahtari, N, S, B, D); // Sınavı Öğrencilere Uyguluyoruz
457
458     for(int ogrenci_ID = 0; ogrenci_ID < N; ogrenci_ID++){ // Öğrenci Sayısı Kadar Döner
459         ogrenci_cevabini_yazdir(ogrenci_cevaplari, ogrenci_ID, S); // Öğrencinin Cevabını Konsola Yazdıran Fonksiyon
460     }
461
462     ogrencileri_puanla(ogrenci_cevaplari, cevap_anahtari, HBN, N, S); // Öğrencilerin Puanlarını Hesaplatıyoruz
463
464     ortalama = sinif_ortalamasini_hesapla(HBN, N); // Sınıf Ortalamasını Çekiyoruz
465     std = standart_sapma_hesapla(ortalama, HBN, N); // Sınıfın Standart Sapmasını Çekiyoruz
466     T_skoru_hesapla(ortalama, HBN, N, std, T_skoru); // Her Öğrenci İçin T-Skoru Hesaplatan Fonksiyonu Çalıştırıyoruz
467
468     printf("\n\nSınıf Ortalamasi: %.2lf, Standart Sapma: %.2lf", ortalama, std); // Sınıfın Bilgilerini Yazıyoruz
469
470     sinif_duzeyi(ortalama); // Sınıf Düzeyini Ortalama Bilgisini Göndererek Ekrana Yazdırıyoruz
471
472     ogrenci_notlari(HBN, N, T_skoru); // Öğrencilerin Puan Tablosunu Konsole Yazdırıyoruz
473
474 }
475

```

ADIM 4 – Sınav Oluşturma Fonksiyonu

Bu fonksiyonu main fonksiyonumuzun daha minimalist olması ve karışıklığa sebebiyet vermemesi için ekledik. Kullanıcıdan alacağımız sınav girdilerini bu fonksiyon içerisinde alıyoruz.

Burada **int dwk** isimli bir değişken oluşturarak kullanıcının belirlediğimiz aralıktaki sayıları girip girmediğini kontrol edip ekrana hata yazısı bastırıyoruz. Eğer kullanıcı örneğin **N** (sınava katılan kişi sayısı) değerini 1 ile 100 aralığında girmezse **do-while** döngüsü daha önce 1 defa çalıştığından ve tekrar çalıştığında **dwk** değeri 1 arttığından ekrana “HATA! En az 1 en fazla 100 değeri girilebilir.” şeklinde hata basacak ve tekrar değer girilmesini isteyecektir. Döngüden her çıktığımızda ise bir diğer döngüde **dwk** değerini kullanabilmemiz için 0 olarak güncelliyoruz.

Aynı şekilde **int S**, **double B**, **double D** değerleri içinde **do-while** döngüleri oluşturup aynı işlemleri kendi veri tipine uygun şekilde yapıyoruz.

Örnek Kod:

```

9 void sinav_olustur(){
10
11     int dwk; // Hata Yazisini Ekrana Bastirmak Icin Kullaniyoruz
12
13     /*
14
15     Sinava Katilan Ogrenci Sayisini Gerekli Sartlari
16     Do While Dongusu Ile Kontrol Ettirerek Kullanicidan Aliyoruz
17
18     */
19
20     do{
21
22         if(dwk > 0){ // Eger dongu sartlarini sagliyorsa yani hata vermis hata cikmesi ekrana yazar.
23             printf("\nHATA! En az 1 en fazla 100 degeri girilebilir.\n");
24         }
25
26         printf("Sinava Katilan Ogrenci Sayisi (Max 100): ");
27         scanf("%d", &N);
28
29         dwk++;
30
31     }while(N < 1 || N > 100);
32
33     dwk = 0; // Her Donguden Sonra Diger Dongude Kullanabilmek Icin Kontrol Degiskenimizi Sifirliyoruz
34

```

Örnek Çıktı:

```
Sinava Katılan Öğrenci Sayısı (Max 100): 5
Sınavdaki Soru Sayısı (Max 100): 10
Sorunun Bos Birakılma İhtimali (0 - 1 Arası Giriniz): 0.1
Sorunun Doğru Yapılma İhtimali (0 - 1 Arası Giriniz): 0.9
```

ADIM 5 – Cevap Anahtarı Üretme Fonksiyonu

Cevap anahtarı üretmek için sistem saatini kullanarak 0 – 4 aralığında rastgele sayılar üretiyoruz. Soru sayımız kadar bir döngü oluşturuyoruz ve `switch` fonksiyonunu kullanarak sayıya denk gelen harfi kontrol edip `char cevap_anahtari` isimli karakter dizimizin içine sorunun indis değerini kullanarak kaydediyoruz.

Örnek Kod:

```
102
103 void cevap_anahtari_uret(char cevap_anahtari[], int S){
104
105     int cevap; // Rastgele Üreteceğimiz Cevaplar İçin Tanımladık
106
107     srand(time(0)); // Sistem Saatine Göre Rastgele Sayı Üretmek İçin Kullanacağız
108
109     for(int i = 0; i < S; i++){
110
111         /* Burada Soruların Rastgele Üretilmiş Cevaplarını cevap_anahtari
112         Dizisine Kaydetmemiz İçin Soru Sayısı Kadar Döngü Başlatıyoruz */
113
114         cevap = rand() % 5; // 0 - 4 Aralığında Rastgele Sayı Üretir
115
116         switch(cevap){ // Burada Rastgele Üretilmiş Sayıya Denk Gelen Harfi cevap_anahtari Dizimize Yerleştiriyoruz
117             case 0:
118                 cevap_anahtari[i] = 'A';
119                 break;
120             case 1:
121                 cevap_anahtari[i] = 'B';
122                 break;
123             case 2:
124                 cevap_anahtari[i] = 'C';
125                 break;
126             case 3:
127                 cevap_anahtari[i] = 'D';
128                 break;
129             default:
130                 cevap_anahtari[i] = 'E';
131         }
132     }
133 }
134
135 }
```

ADIM 6 – Cevap Anahtarı Yazdırma Fonksiyonu

Cevap Anahtarı Üreten fonksiyonumuzu kullanarak oluşturduğumuz cevap anahtarından elde ettiğimiz verileri ekrana yazdırdığımız fonksiyondur.

Örnek Kod:

```
137 void cevap_anahtari_yazdir(char cevap_anahtari[], int S){
138
139     /*
140     Bu Fonksiyon Rastgele Üretilmiş cevap_anahtari Dizimizi
141     Bir Döngü Yardımıyla Konsola Bastırır
142     */
143
144     printf("\nCEVAP ANAHTARI:\n");
145
146     for(int i = 0; i < S; i++){
147         printf("%.3d:%c | ", i+1, cevap_anahtari[i]); // Soruların Numaralarındırmasında Başına 0 Eklenmesi İçin .3d Yazdık
148     }
149
150     printf("\n");
151
152 }
```

Örnek Çıktı:

CEVAP ANAHTARI:

001:E | 002:C | 003:C | 004:D | 005:C | 006:D | 007:C | 008:D | 009:D | 010:B |

ADIM 6 – Sınavı Belirtilen Koşullara Uygun Olarak Öğrencilere Uyguladığımız Fonksiyon

Bu fonksiyonumuz ile her öğrenci için rastgele bir cevap anahtarı oluşturuyoruz. Bu cevap anahtarını oluştururken öncelikle sorunun boş olup olmayacağına boş bırakılma oranına göre karar veriyoruz. Bunu rastgele 1 ile 100 aralığında sayı üretip **if-else** fonksiyonlarımız ile kontrol ettirerek yapıyoruz. Detaylı açıklamalar kod blokları arasındaki yorum satırlarında belirtilmiştir.

Boş olup olmadığını belirledikten sonra aynı işlemi aynı şekilde doğru olup olmadığını belirlemek için uyguluyoruz. Eğer cevap doğru olarsa cevap anahtarından doğru cevabı çekip öğrencinin cevap anahtarına doğru cevabı kaydediyoruz. Eğer cevap yanlış olarsa cevap anahtarında yanlış cevap vermesini sağlamak için sonsuz bir **for** döngüsü oluşturup rastgele cevaplar verdiriyoruz. Cevabı doğru olmayana kadar döngü dönecektir. Yanlış herhangi bir cevap belirlediğinde öğrencinin cevabı olarak o cevap kabul edilecektir.

```
153 void sinavi_uygula(char ogrenci_cevaplari[100][100], char cevap_anahtari[], int N, int S, double B, double D){
154
155     /*
156     Bu Fonksiyonumuz Boş Bırakma, Doğru Cevap Verme Oranlarını
157     Göz Önünde Bulundurarak Öğrenci Sayısı Kadar Rastgele Bir Sınav
158     Oluşturur ve Öğrencilerin Cevaplarını ogrenci_cevaplari Dizisine Kaydeder
159     */
160
161     int cevap, dogruOrani, dogruMu, bosOrani, bosMu; // Bazı Gerekli Değişken Tanımlamalarını Yaptık
162
163     srand(time(0)); // Sistem Saatine Göre Rastgele Sayı Üretmek İçin Kullanacağız
164
165     bosOrani = B*100; // Bu Değişken Double Girdimiz Olan Boş Oranını 100 ile Çarparak Tam Sayı Haline Getiriyor
166     dogruOrani = D*100; // Bu Değişken Double Girdimiz Olan Doğru Oranını 100 ile Çarparak Tam Sayı Haline Getiriyor
167
168     for(int i = 0; i < N; i++){ // Öğrenci Sayısı Kadar Dönen Döngü
169
170         for(int j = 0; j < S; j++){ // Soru Sayısı Kadar Dönen Döngü
171
172             bosMu = rand() % 100 + 1; // 1 - 100 Aralığında Rastgele Bir Sayı Döndürür
173
174             if(bosOrani >= bosMu){ // Eğer Rastgele Dönen Sayı bosOrani Değişkeninden Küçük yada Eşitse Soru Boştur
175
176                 ogrenci_cevaplari[i][j] = 'X';
177
178             }else{ // Değilse Soru Ya Doğrudur Ya Yanlış
179
180                 dogruMu = rand() % 100 + 1; // 1 - 100 Aralığında Rastgele Bir Sayı Döndürür
181
182                 if(dogruOrani >= dogruMu){
183
184                     /*
185                     Eğer Rastgele Dönen Sayı dogruOrani Değişkeninden Küçük yada Eşitse Soru Doğrudur
186                     Bu Durumda cevap_anahtari Dizisinden Soru İndis Numarasına Denk Gelen Cevabı Öğrenciye Kaydediyoruz
187                     */
188
189                     ogrenci_cevaplari[i][j] = cevap_anahtari[j];
190
191                 }else{ // Burası Soruya Yanlış Cevap Verilecekse Çalışır
192
193                     for(;;){ // Soruya Doğru Cevap Verilmeyene Kadar Çalışan Döngü
194
195                         cevap = rand() % 5; // 0 - 4 Aralığında Rastgele Sayı Üretir
196
197                         if(cevap == 0 && cevap_anahtari[j] == 'A' || cevap == 1 && cevap_anahtari[j] == 'B' || cevap == 2 && cevap_anahtari[j] == 'C' || cevap == 3 && cevap_anahtari[j] == 'D' || cevap == 4 && cevap_anahtari[j] == 'E'){
198
199                             /* Eğer Cevap Anahtarındaki Cevap ile Rastgele Çıkan
200                             Cevap Aynıysa Döngünün Bir Sonrakı Adımına Geçer */
201
202                             cevap = rand() % 5; // 0 - 4 Aralığında Rastgele Sayı Üretir
203                             continue; // Sonrakı Adıma Geç
204
205                         }else{ // Eğer Doğru Cevap Denk Gelmemişse
206
207                             switch(cevap){ // Çıkan Cevaba Denk Gelen Harfi Öğrencinin Cevap Anahtarı Dizisine Kaydeder
208                                 case 0:
209                                     ogrenci_cevaplari[i][j] = 'A';
210                                     break;
211                                 case 1:
212                                     ogrenci_cevaplari[i][j] = 'B';
213                                     break;
214                                 case 2:
215                                     ogrenci_cevaplari[i][j] = 'C';
216                                     break;
217                                 case 3:
218                                     ogrenci_cevaplari[i][j] = 'D';
219                                     break;
220                                 default:
221                                     ogrenci_cevaplari[i][j] = 'E';
222
223                             }
224
225                             break; // Sonsuz For Döngüsünü Sonlandırır
226
227                         }
228
229                     }
230
231                 }
232
233             }
234
235         }
236
237     }
238
239 }
```

ADIM 7 – Öğrencinin Cevap Anahtarını Ekrana Yazdıran Fonksiyon

Bu fonksiyon indis numarası gönderilen öğrencinin cevap anahtarını konsola yazdırmaktadır. Tüm öğrencilerin cevap anahtarını konsola bastırmak için `main` fonksiyonumuzun içinde bir `for` döngüsü kullanmıştık. İçerisinde ise her öğrenci için bu fonksiyonu öğrencinin bilgilerini göndererek çağırdık. Konsola yazdırırken öğrenci numaralarının ve soru numaralarının başına 0 eklenmesi için `int` değerlerimizi `%.3d` şeklinde kullanarak yazdırdık.

Örnek Kod:

```
240
241 void ogrenci_cevabini_yazdir(char ogrenci_cevapları[][100], int ogrenci_ID, int S){
242
243     /* Bu Dizi Indis Numarası (ogrenci_ID) Gönderilen
244     Oyuncunun Cevap Anahtarlarını Diziden Alıp Konsola Basar */
245
246     printf("\n%.3d. Öğrencinin Cevapları:\n", ogrenci_ID+1);
247
248     for(int j = 0; j < S; j++){ // Soru Dongusu
249
250         printf("%.3d:%c | ", j+1, ogrenci_cevapları[ogrenci_ID][j]);
251
252     }
253
254 }
255
```

Örnek Çıktı:

```
001. Öğrencinin Cevapları:
001:D | 002:C | 003:D | 004:D | 005:X | 006:E | 007:A | 008:X | 009:D | 010:D |
002. Öğrencinin Cevapları:
001:D | 002:C | 003:D | 004:D | 005:C | 006:X | 007:A | 008:C | 009:X | 010:D |
003. Öğrencinin Cevapları:
001:D | 002:C | 003:D | 004:D | 005:C | 006:E | 007:A | 008:C | 009:D | 010:D |
004. Öğrencinin Cevapları:
001:D | 002:C | 003:D | 004:C | 005:C | 006:E | 007:A | 008:C | 009:D | 010:D |
005. Öğrencinin Cevapları:
001:D | 002:C | 003:D | 004:X | 005:C | 006:E | 007:A | 008:C | 009:D | 010:D |
```

ADIM 8 – Tüm Öğrencilerin Puanlarını (HBN) Hesaplayan Fonksiyon

Öncelikle soru başına düşen puanı 100'ü soru sayısına bölerek hesaplıyoruz. Daha sonra öğrencinin puanını tutacak olan `double` değişkenini, doğru, yanlış ve boş sayısını tutacak olan değişkenleri tanımlıyoruz. Biri öğrencileri biri soruları sıralayacak olan 2 tane iç içe `for` döngüsü kuruyoruz.

Daha sonra öğrencinin verdiği cevaplar ile cevap anahtarındaki cevapları kıyaslıyoruz. Doğru, yanlış ve boş sayılarını koşulları sağladıkları sürece 1'er arttırıyoruz.

Çıkan sayılara göre öğrencinin `HBN` puanını hesaplıyoruz. Hesaplama sırasında 4 yanlışın 1 doğruyu götürmesi gerektiğinden yanlış sayısını 4'e bölüyoruz ve doğru sayımızdan çıkartıp çıkan cevabı soru başına düşen puan ile çarpıyoruz. Eğer öğrencinin `HBN` puanı 0'dan büyükse aynen kaydediyoruz. Eğer 0'dan büyük değilse en az 0 olabileceğinden `HBN` puanını 0 olarak güncelliyoruz. Her öğrenci döngüsünün bitişinden önce doğru, yanlış ve boş sayısını tutan değişkenlerimizi diğer öğrencide kullanacağımız için sıfırlıyoruz.

* Bu fonksiyon ekrana herhangi bir çıktı vermeyecektir. Çıktı için farklı bir fonksiyon kullanılmıştır.

```

255
256 void ogrencileri_puanla(char ogrenci_cevaplari[][100], char cevap_anahtari[], double HBN[], int N, int S){
257
258     double soruPuan = 100/S, ogrenciPuan; // Soru Başına Düşen Puanı Belirliyoruz
259     int dogruSayisi = 0, yanlisSayisi = 0, bosSayisi = 0; // Öğrencinin D/Y/B Değerlerini Saydıracağız
260
261     for(int ogrenci_ID = 0; ogrenci_ID < N; ogrenci_ID++){ // Öğrenci Sayısı Kadar Dönen Döngü
262
263         for(int soru_ID = 0; soru_ID < S; soru_ID++){ // Soru Sayısı Kadar Dönen Döngü
264
265             if(ogrenci_cevaplari[ogrenci_ID][soru_ID] == cevap_anahtari[soru_ID]){ // Eğer Cevap Anahtarıyla Aynıysa
266                 dogruSayisi++; // Doğru Sayısını 1 Arttır
267             }else if(ogrenci_cevaplari[ogrenci_ID][soru_ID] == 'X'){ // Eğer Cevabı Boşsa
268                 bosSayisi++; // Boş Sayısını 1 Arttır
269             }else{ // Eğer Cevabı Yanlışsa
270                 yanlisSayisi++; // Yanlış Sayısını 1 Arttır
271             }
272
273         }
274
275         /*
276         Öğrenci Puanı Hesaplaması Yapıyoruz
277         4 Yanlış 1 Doğruyu Götürmesi Gerektiğinden Yanlış Sayımızı
278         4'e Bölüyoruz ve Doğru Sayımızdan Çıkartıyoruz Çıkan Sonucu
279         Soru Başına Düşen Puan İle Çarpıyoruz
280         */
281
282         ogrenciPuan = (dogruSayisi-(yanlisSayisi/4))*soruPuan;
283
284         if(ogrenciPuan > 0){ // Eğer Öğrenci Puanı 0'dan Büyük Çıkarsa Aynen HBN Dizisinde Kaydediyoruz
285             HBN[ogrenci_ID] = ogrenciPuan;
286         }else{ // Değilse En Az 0 Olması Gerektiğinden 0 Olarak Kaydediyoruz
287             HBN[ogrenci_ID] = 0;
288         }
289
290         /* Tüm Değerleri Bir Sonraki Öğrencide Kullanabilmek İçin Sıfırlıyoruz */
291
292         dogruSayisi = 0;
293         yanlisSayisi = 0;
294         bosSayisi = 0;
295
296     }
297
298 }
299

```

ADIM 9 – Sınıf Ortalaması ve Standart Sapma Hesaplayan Fonksiyonlar

Bu fonksiyonlar proje dosyasında verilen formüllere göre hesaplamalar yapmaktadır.

```

300 double sinif_ortalama_hesapla(double HBN[], int N){
301
302     /* Bu Fonksiyonda Sınıf Ortalamasını Ortalama Formülünü Kullanarak Hesaplıyoruz ve Döndürüyoruz */
303
304     double toplamPuan = 0;
305
306     for(int ogrenci_ID = 0; ogrenci_ID < N; ogrenci_ID++){
307
308         toplamPuan += HBN[ogrenci_ID];
309
310     }
311
312     return toplamPuan/N;
313
314 }
315
316 double standart_sapma_hesapla(double ortalama, double HBN[], int N){
317
318     /* Bu Fonksiyonda Sınıfın Standart Sapmasını Standart Sapma Formülünü Kullanarak Hesaplıyoruz ve Döndürüyoruz */
319
320     double standartSapma = 0, toplam = 0;
321
322     for(int ogrenci_ID = 0; ogrenci_ID < N; ogrenci_ID++){
323
324         toplam += pow((HBN[ogrenci_ID]-ortalama), 2);
325
326     }
327
328     standartSapma = sqrt(toplam / (N-1));
329
330     return standartSapma;
331
332 }
333

```

ADIM 10 – Her Öğrenci İçin T-Skoru Hesaplayan Fonksiyon

Proje dosyasında verilen T-Skoru formülünü kullanarak her bir öğrenci için **for** döngüsü kullanarak ayrı ayrı T-Skoru hesaplaması yapıyoruz. T-Skoru hesaplarken daha önceden hazırladığımız Standart Sapma Fonksiyonu ve Sınıf Ortalaması Fonksiyonu’da bize yardımcı oluyor.

Formül:

$$T\text{-Skor} = 60 + 10 * (\text{Öğrenci Puanı} - \text{Sınıf Ortalaması}) / \text{Standart Sapma}$$

```
333
334 void T_skoru_hesapla(double ortalama, double HBN[], int N, double std, double T_skoru[]){
335
336     /* Öğrenci Sayısı Kadar Bir Dizi Döndürüp T_skoru Formülünü Uyguluyoruz ve T_skoru Dizisine Tek Tek Kaydediyoruz */
337
338     printf("\n");
339
340     for(int ogrenci_ID = 0; ogrenci_ID < N; ogrenci_ID++){
341         T_skoru[ogrenci_ID] = 60 + 10 * (HBN[ogrenci_ID] - ortalama) / std;
342     }
343
344 }
345
```

ADIM 11 – Sınıfın Ortalamasına Göre Sınıf Düzeyi Belirleyen Fonksiyon

Bu fonksiyon bize ortalamaya göre sınıfın düzeyini belirler. Puan aralıkları ise proje dosyasındaki tabloya göre hazırlanmıştır. Sınıf düzeyi belirlendikten sonra o düzeye uygun harf puan aralıklarını da belirliyoruz. Detaylı açıklama kod bloğundaki yorum satırlarında yazmaktadır. Ayrıca bu fonksiyon sınıf düzeyini konsola çıktı olarak verir.

Örnek Kod:

```
345
346 void sinif_duzeyi(double ortalama){
347
348     /* Bu Fonksiyonumuz Sınıfımızın Düzeyini Verilen Tabloya Göre Ortalama Bilgisini Kullanarak Kontrol Ediyor */
349
350     int ekstra; // Tablodaki Aralıklara Göre Artış Yapmak İçin ekstra Değişkenini Tanımladık
351
352     if(ortalama > 80 && ortalama <= 100){
353         printf("\nSınıf Düzeyi: Ustun Basari\n\n");
354         ekstra = 0;
355     }else if(ortalama > 70 && ortalama <= 80){
356         printf("\nSınıf Düzeyi: Mukemmel\n\n");
357         ekstra = 2;
358     }else if(ortalama > 62.5 && ortalama <= 70){
359         printf("\nSınıf Düzeyi: Çok İyi\n\n");
360         ekstra = 4;
361     }else if(ortalama > 57.5 && ortalama <= 62.5){
362         printf("\nSınıf Düzeyi: İyi\n\n");
363         ekstra = 6;
364     }else if(ortalama > 52.5 && ortalama <= 57.5){
365         printf("\nSınıf Düzeyi: Ortanın Ustu\n\n");
366         ekstra = 8;
367     }else if(ortalama > 47.5 && ortalama <= 52.5){
368         printf("\nSınıf Düzeyi: Orta\n\n");
369         ekstra = 10;
370     }else if(ortalama > 42.5 && ortalama <= 47.5){
371         printf("\nSınıf Düzeyi: Zayif\n\n");
372         ekstra = 12;
373     }else{
374         printf("\nSınıf Düzeyi: Kötü\n\n");
375         ekstra = 14;
376     }
377
378     for(int harf_id = 0; harf_id < 9; harf_id++){ // Harf Sayısı Kadar Döndürüyoruz
379
380         duzeyler_min[harf_id] = 27+(harf_id*5)+ekstra; // Bu Harfe Denk Gelen Minimum Puan Değerini duzeyler_min Dizimize Kaydediyoruz
381         duzeyler_max[harf_id] = 32+(harf_id*5)+ekstra; // Bu Harfe Denk Gelen Maksimum Puan Değerini duzeyler_max Dizimize Kaydediyoruz
382
383         if(harf_id == 0){ // Eğer Harf FF ise Minimum Değerini 0 Olarak Güncelliyoruz
384             duzeyler_min[harf_id] = 0;
385         }
386
387         if(harf_id == 8){ // Eğer Harf AA ise Maksimum Değerini 100 Olarak Güncelliyoruz
388             duzeyler_max[harf_id] = 100;
389         }
390     }
391 }
392
393 }
```

Örnek Çıktı:

```
Sinif Ortalamasi: 74.00, Standart Sapma: 13.42
Sinif Duzeyi: Mukemmel
```

ADIM 12 – Harf ID Değerine Göre Harfi Konsola Basan Fonksiyon

Bu fonksiyon öğrencilerin notlarını konsola yazarken hesaplanan harf değerine denk gelen harfi ID numarasına göre tespit edip konsola yazdırmaktadır.

```
94
95 void harflendir(int harf_id){
96
97     /* Bu Fonksiyon Ekrana Gelen Harf ID Değerine Göre Harfi Çıktı Olarak Konsola Yazar */
98
99     if(harf_id == 0){
100         printf("FF");
101     }else if(harf_id == 1){
102         printf("FD");
103     }else if(harf_id == 2){
104         printf("DD");
105     }else if(harf_id == 3){
106         printf("DC");
107     }else if(harf_id == 4){
108         printf("CC");
109     }else if(harf_id == 5){
110         printf("CB");
111     }else if(harf_id == 6){
112         printf("BB");
113     }else if(harf_id == 7){
114         printf("BA");
115     }else{
116         printf("AA");
117     }
118
119 }
120
```

ADIM 13 – Öğrenci Notlarını Konsola Yazdıran Fonksiyon

Bir döngü yardımıyla öğrencileri listeliyoruz. Öğrencinin numaralarının başına 0 eklenmesi için `int` değerlerimizi `%.3d` şeklinde kullanarak yazdırdık. Öğrencinin `HBN` ve `T-Skoru` değerlerini double dizilerinden çekerek `%.2lf` şeklinde kullanarak virgülden sonra 2 hane olmasını sağladık.

```
420
421 void ogrenci_notlari(double HBN[], int N, double T_skoru[]){
422
423     /* Öğrencilerin HBN, T-Skoru ve Harf Notlarını Ekrana Yazdırıyoruz */
424
425     int harf_id = 0;
426
427     for(int ogrenci_ID = 0; ogrenci_ID < N; ogrenci_ID++){
428
429         printf("\n%.3d. Öğrencinin HBN: %.2lf, T-Skoru: %.2lf, Harf Notu: ", ogrenci_ID+1, HBN[ogrenci_ID], T_skoru[ogrenci_ID]);
430
431         while(harf_id < 9){ // Harf Sayımız Kadar Dönen Bir Döngü Oluşturuyoruz
432
433             if(T_skoru[ogrenci_ID] >= duzeyler_min[harf_id] && T_skoru[ogrenci_ID] < duzeyler_max[harf_id]){
434                 /* Eğer Öğrencinin T-Skoru Harf Min ve Harf Max Arasındaysa harflendir Fonksiyonunun İçine Bu Harfi Gönderiyoruz */
435                 harflendir(harf_id);
436                 break; // Şart Sağlandığı İçin Döngüden Çıkıyoruz
437             }
438
439             harf_id++;
440
441         }
442
443         harf_id = 0; // Her Öğrenci İçin Kontrol Yapacağımızdan Sıfırlıyoruz
444
445     }
446
447 }
448
```


Harf notunu hesaplarken ise 9 harf notu bulunduğundan 9 defa dönen bir `while` döngüsü oluşturduk. Daha sonra öğrencinin `T-Skoru` puanı ile o harfin minimum ve maksimum puanlarını kıyasladık eğer bu aralıkta ise harflendir fonksiyonunu çalıştırarak o harfin hangi harf olduğunu ekrana yazdırmasını istedik. Harflendirme yapıldığı için döngüden `break` kullanarak ayrıldık. Her öğrencide bu işlemin tekrarlanması için `harf_id` değişkenimizi sıfırladık.

Tüm Programın Çalıştırılmış Örnek Hali:

```
Sinava Katılan Ogrenci Sayisi (Max 100): 5
Sinavdaki Soru Sayisi (Max 100): 10
Sorunun Bos Birakilma Ihtimali (0 - 1 Arasi Giriniz): 0.1
Sorunun Dogru Yapilma Ihtimali (0 - 1 Arasi Giriniz): 0.9

CEVAP ANAHTARI:
001:A | 002:E | 003:E | 004:A | 005:E | 006:D | 007:E | 008:C | 009:E | 010:A |

001. Ogrencinin Cevaplari:
001:A | 002:E | 003:E | 004:A | 005:E | 006:X | 007:E | 008:X | 009:E | 010:A |
002. Ogrencinin Cevaplari:
001:A | 002:E | 003:E | 004:A | 005:E | 006:D | 007:E | 008:C | 009:E | 010:X |
003. Ogrencinin Cevaplari:
001:A | 002:E | 003:E | 004:A | 005:A | 006:D | 007:E | 008:D | 009:E | 010:A |
004. Ogrencinin Cevaplari:
001:A | 002:E | 003:E | 004:X | 005:E | 006:D | 007:C | 008:X | 009:X | 010:A |
005. Ogrencinin Cevaplari:
001:X | 002:X | 003:C | 004:A | 005:X | 006:D | 007:E | 008:C | 009:E | 010:A |

Sinif Ortalamasi: 74.00, Standart Sapma: 13.42
Sinif Duzeyi: Mukemmel

001. Ogrencinin HBN: 80.00, T-Skoru: 64.47, Harf Notu: BA
002. Ogrencinin HBN: 90.00, T-Skoru: 71.93, Harf Notu: AA
003. Ogrencinin HBN: 80.00, T-Skoru: 64.47, Harf Notu: BA
004. Ogrencinin HBN: 60.00, T-Skoru: 49.57, Harf Notu: CC
005. Ogrencinin HBN: 60.00, T-Skoru: 49.57, Harf Notu: CC

...Program finished with exit code 0
Press ENTER to exit console.□
```