

Algoritmalar ve Programlama Uygulama Dersi Soruları

A Grubu

Soru 1. Klavyeden pozitif bir tamsayı **n** okunacaktır.

Bir sayıyı aşağıdaki özelliklere göre sınıflandıran bir program tasarlaymentınız:

- I. Sayının **asal** olup olmadığını bulunuz.
- II. Sayının **mükemmel sayı** olup olmadığını bulunuz.
 - Mükemmel sayı: Pozitif bölenlerinin (kendisi hariç) toplamı kendisine eşit olan sayı (örneğin: $6 \rightarrow 1 + 2 + 3 = 6$).
- III. Program tüm bu kontrolleri **if-else yapıları** ve **döngüler** kullanarak yapmalıdır.

İstenilenler:

- a) Uygun kodlamayı C dilinde yazınız.
- b) Algoritmanın adımlarını (sözde kod) yazınız.
- c) Asal kontrolü, bölenlerin toplamı için hangi döngü türlerini (for / while / do-while) seçtiğinizi gerekçesiyle açıklayınız.
- d) Programın olası **3 farklı çıktı örneği** veriniz (örneğin: bir asal-mükemmel olmayan, bir mükemmel).

Algoritmalar ve Programlama Uygulama Dersi Soruları

A Grubu

Soru 2. Bir zar oyunu tasarlayınız. Oyunda **kullanıcı** ile **bilgisayar** karşılıklı oynayacaktır.

Oyun kuralları:

I. Her turda:

- Oyuncu **2 adet zar** atar.
- Bilgisayar da **2 adet zar** atar.
- Oyuncunun zarları toplanır → oyuncuToplam
- Bilgisayarın zarları toplanır → pcToplam

II. Karşılaştırma:

- Eğer oyuncuToplam > pcToplam ise → bu turu **oyuncu** kazanır ve oyuncunun puanı 1 artar.
- Eğer pcToplam > oyuncuToplam ise → bu turu **bilgisayar** kazanır ve bilgisayarın puanı 1 artar.
- Eşitse → tur **berabere** biter, kimse puan almaz.

III. Oyun, oyuncu veya bilgisayar **5 puana ilk ulaşan** kişi olana kadar devam eder.

IV. Her turda:

- Atılan zarların değerleri,
- Toplamlar,
- “Turu kim kazandı?” bilgisi,
- Güncel skor (Oyuncu: X – Bilgisayar: Y) ekrana yazdırılmalıdır.

V. Oyun bittiğinde:

- Hangi tarafın oyunu kazandığı ekrana yazdırılmalıdır.

VI. Oyun sonunda kullanıcıya:

- “Tekrar oynamak istiyor musunuz? (1 = Evet, 0 = Hayır)” sorusu sorulmalı,
- Kullanıcı 1 girerse oyun **yeniden başlamalı**, 0 girerse program sonlanmalıdır.

Algoritmalar ve Programlama Uygulama Dersi Soruları

A Grubu

Soru 3. Aşağıdaki işlemleri **hiç dizi kullanmadan**, sadece tamsayı aritmetiği ve özyineleme ile yapınız. Kullanıcıdan pozitif bir tamsayı **n** alınacaktır.

I. Basamak Toplamı (Recursive)

Aşağıdaki prototipe sahip fonksiyonu yazınız:

```
int basamakToplami(int n);
```

- n sayısının basamakları toplamını **özyinelemeli** olarak hesaplasın.
- Temel durum ($n < 10$) ve recursive durum için kullandığınız formülü açıklayınız.

II. Tersini Yazdırma (Recursive)

Aşağıdaki prototipe sahip fonksiyonu yazınız:

```
void tersiniYaz(int n);
```

- Sayının rakamlarını **tersten** ekrana yazdıran özyinelemeli bir fonksiyon tasarılayınız.
- Örnek: $n = 1234$ için çıktı: 4321
- İşlem sırasında **ek bir değişkene tüm sayıyı ters çevirip atmak zorunda değilsiniz**, doğrudan ekrana yazdırabilirsiniz.

III. Palindrom Kontrolü (Recursive Yardımcı Fonksiyon ile)

Palindrom: Tersi kendisine eşit olan sayıdır.

Örneğin: 121, 1331, 5 palindromdur; 123 değildir.

- n sayısının palindrom olup olmadığını kontrol eden özyinelemeli bir çözüm tasarılayınız.
- Bunun için isterseniz şu tarz yardımcı fonksiyonlar tanımlayabilirsiniz:

```
int basamakSayisi(int n);
```

```
int tersOlustur(int n);
```

```
int palindromMu(int n);
```

İstenilenler:

- Uygun kodlamayı C dilinde yazınız.
- Sözde kodunu yazınız.
- Programı kendi girdilerinizle test ediniz.

Algoritmalar ve Programlama Uygulama Dersi Soruları

A Grubu

Soru 4. Kullanıcıdan iki pozitif tamsayı **a** ve **b** alınsın.

- I. *ebobIteratif* isimli fonksiyon ile EBOB(a, b) değerini **iteratif** bir yöntemle (döngü kullanarak),
- II. *ebobRekursif* isimli fonksiyon ile EBOB(a, b) değerini **özyinelemeli** olarak (örneğin Öklid algoritması),
- III. *usAl* isimli fonksiyon ile a^b değerini **döngü** ile hesaplayınız.

Fonksiyon prototipleri:

```
int ebobIteratif(int a, int b);
```

```
int ebobRekursif(int a, int b);
```

```
long long usAl(int taban, int us);
```

İstenilenler:

- a) *ebobIteratif* için basit bir döngü tabanlı yöntem tasarlaymentınız (örneğin 1'den min(a,b)'ye kadar bölen deneme veya daha verimli bir döngü).
- b) *ebobRekursif* için:
 - o Recursive adımı nasıl tanımladığınızı yazınız.
- c) main fonksiyonunda:
 - o Kullanıcıdan a ve b alınsın,
 - o Her iki EBOB fonksiyonu çağrılsın, sonuçlar karşılaştırılsın,
 - o Ardından *usAl(a, b)* sonucu da ekrana yazdırılsın.