

İSTANBUL ÜNİVERSİTESİ AUEF - Bilgisayar Programcılığı

ALGORİTMA VE PROGRAMLAMAYA GİRİŞ

(1-4 Ünite VİZE) 2022 I (1-8 Ünite FİNAL) 2022

1-TEMEL KAVRAMLAR Ünite Soruları

Soru-1 : Aşağıdakilerden hangisi algoritmayı ifade etmek için kullanılan yöntemlerden değildir?

- (•) - Söзде kod (•) - **Sözlü kod** (•) - Akış diyagramı (•) - Kaba kod (•) - Pseudo kod

Soru-2 : Hangisi büyük O notasyonu ile ölçülen bir kavramdır?

- (•) - çalışma süresi (•) - işlem miktarı (•) - algoritma türü (•) - **zaman karmaşıklığı** (•) - değişken sayısı

Soru-3 : Aşağıdaki hangisi veritabanını ifadede kullanılır?



Soru-4 : Karar yapılarını ifade etmek için hangi eleman kullanılır?



Soru-5 : Aşağıdakilerden hangisi işlemleri veya eylemleri ifade etmek için kullanılır?



Soru-6 : Akış diyagramının hangi elemanında birden fazla çıktı olabilir?

- (•) - **Karar yapıları** (•) – Fonksiyonlar (•) – Başlangıç (•) – İşlem (•) - Hiçbiri

Soru-7 : Bir akış diyagramında birden fazla başlangıç ve bitiş elemanı kullanılabilir mi?

- (•) - Kullanılabilir. **(•) - Kullanılmaz.** (•) - Döngü yapısı ile birlikte kullanılabilir.
(•) - Koşullu yapıda kullanılabilir. (•) - Alt programlarda kullanılabilir.

Soru-8 : Aşağıdakilerden hangisi bir algoritmada olması gereken özelliklerinden değildir?

- (•) - Sonlu olma (•) – Etkinlik **(•) – Verimlilik** (•) – Kesinlik (•) - Doğruluk

Soru-9 : Aşağıdakilerden hangisi bir algoritmada olabilecek kontrol yapılarından değildir?

- (•) - Sıralı yapı (•) - Seçimli yapı (•) - Tekrarlı yapı (•) - Koşullu yapı **(•) - Sonlu yapı**

Soru 10: Aşağıdakilerden hangisi ile ifade edilen bir algoritma diğerlerine göre daha hızlı çalışır?

- (•) - $O(n^3)$ (•) - $O(n \log n)$ **(•) - $O(1)$** (•) - $O(\log n)$ (•) - $O(n)$

2-PROGRAMLAMAYA GİRİŞ Ünite Soruları

Soru-1 : Aşağıdakilerden hangisi tam sayı veri tiplerinden değildir?

- (•) – sbyte (•) – short (•) – int **(•) – sint** (•) - ulong

Soru-2 : Aşağıdakilerden hangisi karakter veri tipindedir?

- (•) – Char** (•) – strng (•) – byte (•) – letter (•) - stg

Soru-3 : Aşağıdakilerden hangisi aritmetik işlem operatörü değildir?

- (•) – toplama (•) – çıkarma **(•) - kesişim kümesi oluşturma** (•) - mod alma (•) - üs alma

Soru-4 : Aşağıdakilerden hangisi mantıksal işlem operatörlerindendir?

- (•) - AND NOT (•) - OR AND **(•) – NOT** (•) - OR NOT (•) - NOT OR

Soru-5 : Aşağıdakilerden hangisi karşılaştırma operatörüdür?

- (•) – Eşittir (•) - küçük veya büyüktür (•) – true (•) – false **(•) - eştir**

Soru-6 : Karakterler hangi tablo referans alınarak tanımlanmaktadır?

- (•) - char Tablosu (•) - akış tablosu **(•) – ASCII** (•) – AXCI (•) - Karakter tablosu

Soru-7 : Aşağıdakilerden hangisi sayısal ve karakter tipindeki verilerin dışındaki veri tiplerindendir?

- (•) – string **(•) – boolean** (•) – float (•) – short (•) – char

Soru-8 : Basit algoritma yapısında işlemler birbirini nasıl takip eder?

- (•) - Ardışık olarak** (•) - Bazı adımları atlayarak (•) - Bazı adımları tekrarlayarak
(•) - Sondan başa doğru (•) - Hiçbiri

Soru-9 : AND mantıksal operatöründe sonucun doğru (TRUE) döndürülmesi için aşağıdakilerden hangi koşulun sağlanması gerekir?

- (•) - Durumlardan birinin doğru olması (•) - Durumlardan en az birinin doğru olması
(•) - Her iki durumun doğru olması (•) - Her iki durumun yanlış olması (•) – Hiçbiri

Soru 10: Bir algoritmada artışın ifade edilmesi için aşağıdakilerden hangisi kullanılmaz?

- (•)- sayaç = sayaç + 1 (•)- c = ++a (•)- b = a++ (•)- a = a + 1 **(•)- a++-1**

3-KARAR YAPILARI Ünite Soruları

Soru-1 : “Eğer yemek yediysen ilacını al.” cümlesi ile ilgili aşağıdakilerden hangisi yanlıştır?

- (•) - Tek bir koşulun kontrolü yapılır. (•) - Koşulun doğru olması durumunda sadece bir işlem yapılır.
(•) - Koşulun yanlış olduğu durumda ne yapılacağı bellidir. (•) - Koşulun yanlış olduğu durumda hangi işlemin yapılacağı belli değildir. (•) - İlaç alma eylemi bir koşula bağlıdır.

Soru-2 : Aşağıdaki boşluğa gelmesi gereken doğru kelime hangisidir?

“Kodun okunurluğunun artması amacıyla koşullu yapıların yazımında kodların _____ yazılması önerilir.”

- (•) – birleşik **(•) – girintili** (•) - ardışık olarak (•) - iç içe (•) - sırayla

Soru-3 : Koşullu yapıların kullanım amacı ile ilgili olarak aşağıdaki cümlelerden hangisi yanlıştır?

- (•) - Bu yapılar bazı adımların atlanmasını sağlar. (•) - Bu yapılar bir seçim yapılmasını sağlar.
(•) - Bu yapılar ardışık işlem yapılmasını zorunlu kılar.
 (•) - Bu yapılar bir durumun kontrol edilmesini sağlar. (•) - Bu yapılar sayesinde istenen bir koşul sınanır.

Soru-4 : Bir öğrencinin dersten geçme durumunu ekrana yazdıran kod parçasında, dersten kalındıysa ekrana mesaj döndürülür, kalınmadıysa program durur. Bu programda kaç durumun kontrolü yapılır?

- (•) – 1** (•) – 2 (•) – 3 (•) – 4 (•) – 5

Soru-5 : Öğrencinin notu 50 ise CC, notu 70 ise BB, 90 ise AA değerlerini döndüren bir programla ilgili olarak aşağıdakilerden hangisi doğrudur?

- (•) - Programın çıktısı yoktur. **(•) - Birden fazla koşul kontrolü yapılır.** (•) - Öğrenci notu yanlış girilmiştir.
 (•) - Farklı durumların hepsi aynı sonucu döndürür. (•) - Öğrencinin notu boolean tipinde bir değerdir.

Soru-6 : Aşağıdaki önermede boş bırakılan yer koşullu bir yapıyı ifade etmek için hangi kelime gelmelidir ? “_____ paketin gelmediyse firmayı arayıp kargonun nerede olduğunu sor.”

- (•) – Elseif (•) - Çok seçimli yapı (•) – Beklediğin (•) – İsteddiğin **(•) - Eğer**

Soru-7 : Bir programda birden fazla if deyimi bir arada kullanıldığında hangi deyim if deyimi ile aynı sayıda kullanılmalıdır?

- (•) - End if** (•) – elseif (•) – else (•) – select (•) - Hiçbiri

Soru-8 : İç içe seçimli yapı ile ilgili aşağıdakilerden hangisi kontrol edilir?

- (•) - En dıştaki koşul içeridekine bağlı olarak çalışır. (•) - Koşulların mutlaka sağlanması gerekir.
 (•) - Koşulların ikisi de çalışmaz. (•) - Koşulların ikisi de aynı anda çalışır.
(•) - En dıştaki koşula bağlı olarak içerideki koşul çalışır veya çalışmaz.

Soru-9 : Girilen sayının 100’e eşit olduğu durumda “Sayı doğru” eşit olmadığı durumda “Sayı yanlış” mesajını döndüren bir programın sözde kodu hangisidir?

- (•) - EĞER sayı = 100 ise (•) - EĞER sayı <> 100 ise

YAZ (“Sayı doğru”)

YAZ (“Sayı yanlış”)

(•) - EĞER sayı = 100 ise

(•) - EĞER sayı <> 100 ise

YAZ (“Sayı doğru”)

YAZ (“Sayı doğru”)

EĞER sayı = 100 değilse

EĞER sayı = 100 değilse

YAZ (“Sayı yanlış”)

YAZ (“Sayı doğru”)

- (•) - EĞER sayı - 100 ise

YAZ (“Sayı doğrudur”)

EĞER sayı = 100 değilse

YAZ (“Sayı yanlıştır”)

Soru 10: Aşağıdaki seçeneklerden hangisi “ders_adi” değişkeninin aldığı değerin “algoritma” olduğu koşulun kontrolünü sağlayan yapı hangisidir?

- (•) - ders_adi <> algoritma (•) - ders_adi = algoritma (•) - Dersimin_adi = “algoritma”
 (•) - **ders_adi = “algoritma”** (•) - Ders & “algoritma”

4-DÖNGÜLER Ünite Soruları

Soru-1 : do while...loop döngüsünde kontrol hangi aşamada yapılır?

- (•) - **İlgili kod bloğunun başında** (•) - İlgili kod bloğunun sonunda (•) - Program durduktan sonra
 (•) - Başla komutundan önce (•) - Hiçbiri

Soru-2 : do loop...while döngüsünde kontrol hangi aşamada yapılır?

- (•) - İlgili kod bloğunun başında (•) - **İlgili kod bloğunun sonunda** (•) - Hiçbiri
 (•) - Program durduktan sonra (•) - Başla komutundan sonra

Soru-3 : do while...loop ve do loop...while döngü yapıları arasındaki temel fark nedir?

- (•) - Sadece yazım şekillerinin farklı olması (•) - Farklı dillerde yazılmış olmaları (•) – Hiçbiri
 (•) - Tekrarlanan işlemin uzun olması (•) - **Birinde tekrarlanmak istenen işlemin en az bir kez yürütülmesi**

Soru-4 : Sayı 100 iken aşağıdaki kod çalışmaktadır. Bu kodda yapılan hata nedir?

DO WHILE (sayı >= 0)

YAZ (sayı)

LOOP

- (•) - Yazım hatası yapılmıştır. (•) - **Döngü sonlanmamaktadır.** (•) - Tekrarlanan işlem sonuç vermez.
 (•) - Geçersiz bir sayı girilmiştir. (•) - Hiçbiri

Soru-5 : 0 - 10 arasındaki sayıların toplamını do while...loop döngüsü kullanarak hesaplamak için hangi kod doğru yazılmıştır? (sayı = 0, toplam = 0 başlangıçta tanımlanmış varsayılır.)

- (•) - DO LOOP (sayı < 10) (•) - DO WHILE (sayı < 10)

toplam = toplam + sayı

toplam = toplam + sayı

sayı = sayı ++

sayı = sayı +

WHILE

LOOP

- (•) - **DO WHILE (sayı < 10)**

- (•) - DO WHILE (sayı < 10)

toplam = toplam + sayı

toplam = toplam + sayı

sayı = sayı ++

sayı = sayı + sayaç

LOOP

LOOP

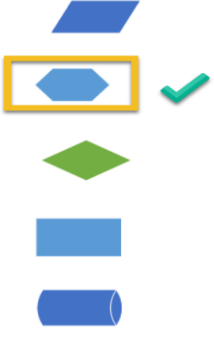
- (•) - DO LOOP (sayı < 10)

toplam = toplam + sayı

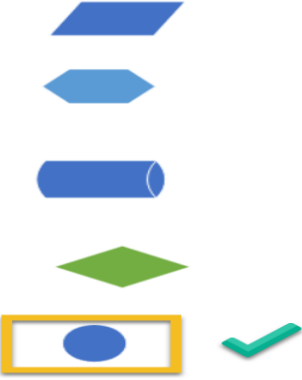
sayı + 1 = sayı

WHILE

Soru-6 : Aşağıdaki kontrollerden hangisi döngülerin bir türünü ifadede kullanılır?



Soru-7 : Bir for döngüsünde sayaç değişkeninin artması gerektiğini ifade etmek için hangi eleman kullanılır?



Soru-8 : 10-0 arasındaki sayıları for döngüsü kullanarak azalarak yazdırılmak istendiğinde aşağıdakilerden hangisi bu yapı ile ilgili doğru bir ifade olur?

- (•) - Başlangıç değeri 0 olmalıdır. (•) - Bitiş değeri 10 olmalıdır.
- (•) - Sayaç birer birer arttırılmalıdır. (•) - Sayaç birer birer azaltılmalıdır.
- (•) - Sayaç olmasına gerek yoktur.

Soru-9 : f or döngüsü ile ilgili aşağıdakilerden hangisi yanlıştır?

- (•) - Başlangıç değeri belirlenmelidir. (•)- Bitiş değeri belirlenmelidir.
- (•) - Akış diyagramında sayaç ayrı bir işlem elemanı olarak gösterilebilir.
- (•) - Akış diyagramında altıgen eleman ile gösterilebilir. (•) - Artış miktarı mutlaka belirtilmelidir.

Soru 10: Bir müşteri listesindeki müşterilerin numaralarını for döngüsü kullanarak yazdıran kod ile ilgili hangisi doğrudur?

- (•)-Döngü, listedeki son müşteri işlendikten sonra durur. (•)-Müşteri listesindeki istenen elemanlar karakter tipindedir.
- (•) - Döngü sonlanmaz.(•) - Listedeki elemanlar yazdırılamaz. (•) - Müşteri adı indis numarasını ifade eder.

5-ALT PROGRAMLAR FONKSİYONLAR Ünite Soruları

Soru 1: Bir programdaki ana program nasıl adlandırılır?

- (•)- void (•)- main (•)-proc (•)-mainprog (•)-static void

Soru 2: Geri değer döndürmeyen alt programlar aşağıdakilerden hangisi gibi ifade edilemez?

- (•)- void (•)-procedure (•)-proc (•)-func (•)-void Programİsmi

Soru 3: Geriye değer döndüren alt programlar aşağıdakilerden hangisi gibi adlandırılır?

- (•)- void (•)- procedure (•)-proc (•)-function (•)-return

Soru 4: Geriye değer döndüren alt programlar aşağıdakilerden hangisi ile çağırılır?

- (•)-call (•)-Uygun değişkene eşitlenerek (•)-main (•)-void metodu ile (•)-Hiçbiri

Soru 5: Kendini çağıran programlara ne ad verilir?

- (•)- void (•)-Geri döndürmeyen alt program (•)-Tekrarlı ana program
(•)-Geri değer döndüren alt program **(•)- Özyneli fonksiyon**

Soru 6: Değişkenlerin ismi parametre listesinde tanımlandıktan sonra alt program içerisinde tekrar kullanılabilir mi?

- (•)- Evet **(•)-Hayır** (•)-Koşula göre değişir. (•)-Döngü varsa kullanılır. (•)-Hiçbiri

Soru 7: Geri değer döndüren alt programdan dönen paramtrenin cinsi ana programda o fonksiyonu çağırırken kullanılan parametre türüyle uyuşmazsa ne olur?

- (•)-Program sorunsuz çalışır. (•)-Sadece alt program çalışır. (•)-Program sonsuz döngüye girer.
(•)-Hata alınır. (•)-Hiçbir şey olmaz.

Soru 8: Metotların içerisinde tanımlanan değişkenler metot dışında kullanılabilir mi?

- (•)-Kullanılabilirler. **(•)-Kullanılamazlar.** (•)-Duruma göre kullanılabilir.
(•)-Özyine fonksiyonsa kullanılabilir. (•)-Hiçbiri

Soru 9: Özyneli fonksiyonlar nasıl nitelendirilebilir?

- (•)- Kaba (•)-Verimsiz (•)-Çok uzun **(•)-Öz** (•)-Hiçbiri

Soru 10: Farklı programlama dillerinde ana program aşağıdakilerden hangisi ile ifade edilmez?

- (•)-Sub main (•)-Main **(•)-Main Code** (•)-Void main (•)-Hiçbiri

6-DİZİLER Ünite Soruları

Soru 1: Tek boyutlu dizilerde indis değerini ifade etmek için sıklıkla hangi harf kullanılır?

- (•)- a **(•)-i** (•)-l (•)-z (•)-x

Soru 2: İki boyutlu dizilerde ikinci indis değerini ifade etmek için sıklıkla hangi harf kullanılır?

- (•)- a (•)-p (•)-l **(•)-j** (•)-x

Soru 3: Aşağıdakilerden hangisi dizilerin indislerini ifade etmek için kullanılabilir?

- (•)- Sadece parantez (•)- Sadece köşeli parantez (•)-Sadece tırnak işareti
(•)-Parantez veya köşeli parantez (•)- Süslü parantez

Soru 4: Dizilerin indis numarası ile ilgili aşağıdakilerden hangisi doğrudur?

- (•)- İndis numarası mutlaka 0'dan başlar. (•)-İndis numarası mutlaka 1'den başlar.
(•)-İndis başlangıç numarası dilden dile farklılık gösterir. (•)-İndis numarası x'ten başlar.
(•)- İndis numarası string tipinde bir değerdir.

Soru 5: Dizilerin zaman karmaşıklığı ile ilgili olarak aşağıdakilerden hangisi doğrudur?

- (•)- Dizilerin zaman karmaşıklığı çok azdır. (•)- Dizilerin zaman karmaşıklığı ölçülemez.
(•)-Tek boyutlu dizilerin zaman karmaşıklığı iki boyutlu dizilere göre daha fazladır.
(•)-İki boyutlu dizilerin zaman karmaşıklığı tek boyutlu dizilere göre daha fazladır.
(•)-Tek ve iki boyutlu dizilerin karmaşıklığı her zaman aynıdır.

Soru 6: Diziler ile ilgili olarak aşağıdakilerden hangisi yanlıştır?

- (•)-Dizilerde string tipinde değerler tutulabilir. (•)- Dizilerde char tipinde değerler tutulabilir.
(•)-Dizilerde sayısal değerler tutulabilir. **(•)-Bir dizinin elemanları aynı veri tipinde olmak zorunda değildir.**
(•)-Bir dizinin eleman sayısı programcı tarafından belirlenir.

Soru 7: Dizilerin indis numarası ile ilgili aşağıdakilerden hangisi doğrudur?

- (•)- İndis numarası mutlaka 0'dan başlar. (•)-İndis numarası mutlaka 1'den başlar.
(•)-İndis başlangıç numarası dilden dile farklılık gösterir. (•)-İndis numarası x'ten başlar.
(•)-İndis numarası string tipinde bir değerdir.

Soru 8: Aşağıdakilerden hangisi iki boyutlu diziler ile ilgili yanlıştır?

- (•)-Birden fazla tek boyutlu dizinin bir araya gelmesi ile oluşur. (•)- Satır ve sütunlardan oluşur.
 (•)-İki boyut iki ayrı indis ile ifade edilir. (•)- Elemanları sadece sayısal tipte olabilir.
 (•)-Bir dizinin elemanları aynı veri tipinde olmak zorunda değildir.

Soru 9: İki boyutlu dizinin oluşturulması ile ilgili olarak hangisi doğrudur?

- (•)-İki boyut i indisi ile ifade edilir. (•)-Seçimli yapı kullanılarak oluşturulması gerekir.
 (•)-İki boyutlu dizide sadece sayısal değerler tutulabilir. (•)-Bir for döngüsü kullanılarak oluşturulması gerekir.
 (•)-İki for döngüsü kullanılarak oluşturulabilir.

Soru 10: 1) İki boyutlu dizilerin zaman karmaşıklığı hangisi ile ifade edilir?

- (•)- $O(\log n)$ (•)- $O(n)$ (•)- $O(n \log n)$ (•)- $O(n^2)$ (•)- $O(n^3)$

7-ARAMA VE SIRALAMA ALGORİTMALARI Ünite Soruları

Soru 1: Aşağıdakilerden hangisi bilinen bir arama algoritmasıdır?

- (•)- Çapraz arama (•)-İkili arama (•)-Arama algoritması (•)-Eleman arama (•)-Hiçbiri

Soru 2: Aşağıdakilerden hangisi bilinen bir arama algoritmasıdır?

- (•)-Çapraz arama (•)-Tek tek arama (•)-Arama algoritması (•)-Doğrusal arama (•)-Hiçbiri

Soru 3: Aşağıdakilerden hangisi karşılaştırma temelli olarak çalışan bir sıralama algoritmasıdır?

- (•)- Hızlı sıralama (•)- Birleştirmeli sıralama (•)-İkili sıralama (•)-Kabarcık sıralama (•)-Hiçbiri

Soru 4: Aşağıdakilerden hangisi karşılaştırma temelli olarak çalışan bir sıralama algoritmasıdır?

- (•)- Hızlı sıralama (•)-Yerleştirmeli sıralama (•)-İkili sıralama (•)-Doğrusal sıralama (•)-Hiçbiri

Soru 5: Aşağıdakilerden hangisi böl ve fethet mantığına göre çalışan bir sıralama algoritmasıdır?

- (•)- Hızlı sıralama (•)-Yerleştirmeli sıralama (•)-Birleşik sıralama (•)-Doğrusal sıralama (•)-Hiçbiri

Soru 6: Aşağıdakilerden hangisi böl ve fethet mantığına göre çalışan bir sıralama algoritmasıdır?

- (•)- Lineer sıralama (•)- Yerleştirmeli sıralama (•)- Birleşik sıralama (•)-Birleştirmeli sıralama (•)-Hiçbiri

Soru 7: İkili arama algoritmasının ön şartı nedir?

- (•)- Lineer sıralamanın uygulanmış olması gerekir. (•)-Dizide aranan elemanın olması gerekir.
 (•)- Herhangi bir ön şartı yoktur. (•)-Dizinin küçükten büyüğe sıralanmış olması gerekir. (•)-Hiçbiri

Soru 8: Böl ve fethet algoritmaları ile ilgili hangisi yanlıştır?

- (•)- Problem küçük alt problemlere ayrılarak çözülür. (•)-Özyineli olarak çalışan algoritmalarıdır.
 (•)-Sıralama tek seferde yapılır. (•)-Birden fazla algoritma bu aileye aittir (•)-Sıralanan dizi alt dizilere ayrılır.

Soru 9: 1) Aşağıdakilerden hangisi ile ifade edilen zaman karmaşıklığı en büyüktür?

☐ $O(\log n)$

☐ $O(n)$

☐ $O(n \log n)$

☐ $O(n^2)$

☒ $O(n^3)$ 

Soru 10: Aşağıdaki sıralama algoritmaları birbiri ile karşılaştırıldığında hangisi daha hızlıdır?

- (•)-Doğrusal sıralama (•)-Hızlı sıralama (•)-Birleştirmeli sıralama (•)-Kabarcık sıralama (•)-Yerleştirmeli sıralama

8-GENEL UYGULAMALAR Ünite Sonu Uygulama Soruları

a) Haftalık normal çalışma saati süresi 40 saat ve aldığı ücret 3000 lira olan bir çalışanın mesai yaptığı her saat başına ekstra 50 lira kazandığı biliniyor. Kullanıcı tarafından girilen mesai saatine göre çalışanın alması gereken haftalık toplam ücreti hesaplayıp ekrana yazdıran algoritmayı yazın ve akış diyagramını çizin.

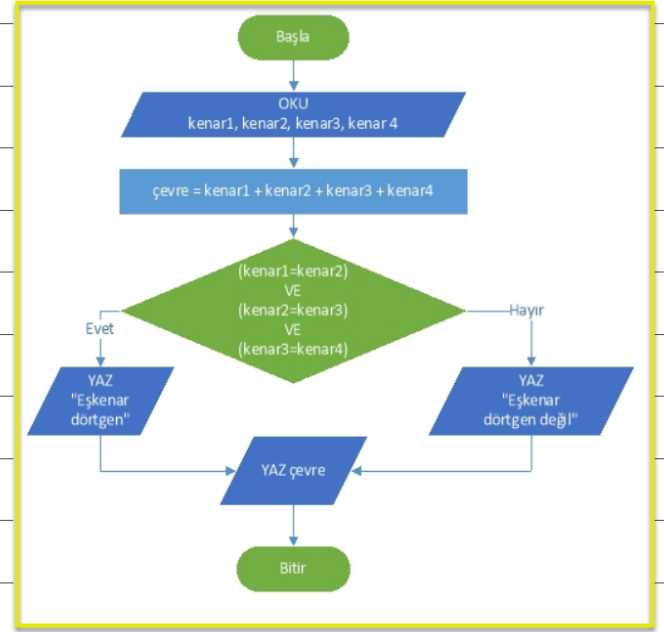
BAŞLA	<pre> graph TD Start([Başla]) --> Read[/mesai saati OKU/] Read --> Decision{mesai saati > 40} Decision -- Evet --> ExtraPay1[fazla mesai = 40 - mesai saati] Decision -- Hayır --> ExtraPay2[fazla mesai = 0] ExtraPay1 --> ExtraPayCalc[ekstra ücret = 50 * mesai saati] ExtraPay2 --> ExtraPayCalc ExtraPayCalc --> TotalPayCalc[total ücret = ekstra ücret + 3000] TotalPayCalc --> Print[/toplam ücret/] Print --> End([Bitir]) </pre>
OKU mesai saati	
IF mesai saati > 40 THEN	
fazla mesai = 40 – mesai saati	
ELSE	
fazla mesai = 0	
END IF	
ekstra ücret = 50 * fazla mesai	
toplam ücret = ekstra ücret + 3000	
YAZ toplam ücret	
BİTİR	

b) Fiyatı 100.000 TL ve yukarı evler için ödenecek olan vergi oranı %20, fiyatı 50.000 ile 100.000 TL arası evler için vergi oranı %15, fiyatı 50.000 TL'den az olanlar içinse vergi oranı %10 olarak belirlenmiştir. Buna göre kullanıcı tarafından girilen fiyata göre ödenmesi gereken vergi miktarını hesaplayan algoritmayı yazınız ve akış diyagramını çizin.

BAŞLA	<pre> graph TD Start([Başla]) --> Read[/Fiyat girin./] Read --> Decision1{fiyat >= 100000} Decision1 -- Evet --> Tax1[verg = fiyat * 0,2] Decision1 -- Hayır --> Decision2{fiyat >= 50000} Decision2 -- Evet --> Tax2[verg = fiyat * 0,15] Decision2 -- Hayır --> Tax3[verg = fiyat * 0,1] Tax1 --> Print[/YAZ vergi/] Tax2 --> Print Tax3 --> Print Print --> End([Bitir]) </pre>
OKU fiyat	
IF fiyat >= 100000 THEN	
verg = fiyat * 0,2	
ELSE IF fiyat >= 50000	
verg = fiyat * 0,15	
ELSE	
verg = fiyat * 0,1	
END IF	
YAZ (vergi)	
BİTİR	

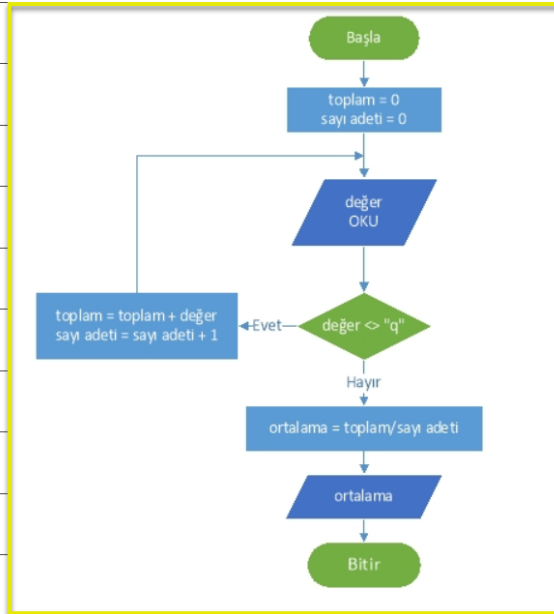
- c) Kenar ölçüleri verilen bir dörtgenin eşkenar dörtgen olup olmadığını hesaplayan ve çevre uzunluğunu bulan algoritmayı yazınız ve akış diyagramını çizin.

BAŞLA
OKU kenar1, kenar2, kenar3, kenar4
$\text{çevre} = \text{kenar1} + \text{kenar2} + \text{kenar3} + \text{kenar4}$
IF (kenar1=kenar2) AND (kenar2=kenar3) AND (kenar3=kenar4) THEN
YAZ ("Eşkenar dörtgen")
ELSE
YAZ ("Eşkenar dörtgen değil")
END IF
YAZ (çevre)
BİTİR



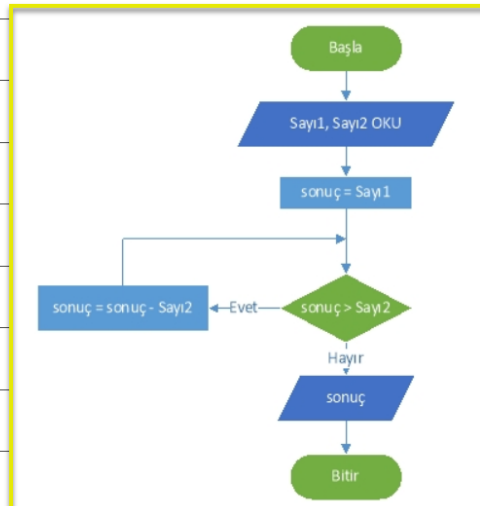
- d) Kullanıcı "q" değerini girine kadar sayı alıp "q" değerini girince bu sayıların ortalamasını alan algoritmayı yazın ve akış diyagramını çizin.

BAŞLA
toplam = 0, sayı adedi = 0, değer = 0
OKU değer
DO WHILE değer <> "q"
toplam = toplam + değer
sayı adedi = sayı adedi + 1
OKU değer
LOOP
ortalama = toplam/sayı adedi
YAZ (ortalama)
BİTİR



- e) '%' operatörünü kullanmadan verilen 2 sayının modunu alan uygulamanın algoritmasını yazın ve akış diyagramını çizin.

BAŞLA
OKU sayı1, sayı2
sonuç = sayı1
DO WHILE sonuç > sayı2
sonuç = sonuç - sayı2
LOOP
YAZ sonuç
BİTİR



f) Girilen harf "q" değilse kullanıcı tarafından girilen harfleri ekranda gösteren, girilen harf "q" ise programdan çıkış yapan algoritmayı yazınız ve akış diyagramını çiziniz.

BAŞLA

OKU harf

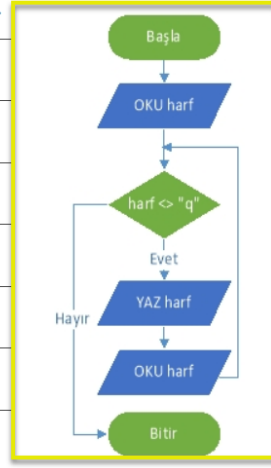
DO WHILE harf <> "q"

YAZ (harf)

OKU harf

LOOP

BİTİR



g) 1 ile 50 arasındaki tam sayılardan tek sayıların toplamı ile çift sayıların toplamının farkının negatif mi, pozitif mi olduğunu bulan programın algoritma ve akış diyagramını çiziniz.

BAŞLA

tek = 0, çift = 0

FOR i = 1 TO 50 Step 2

tek = tek + i

çift = çift + (i + 1)

NEXT i

IF ((tek - çift) < 0) THEN

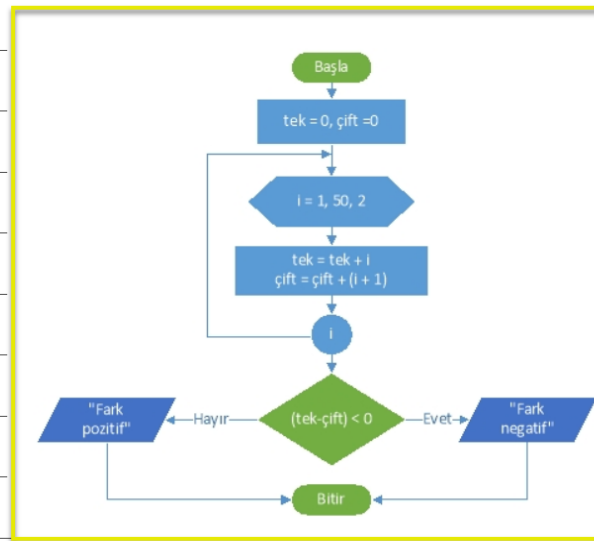
YAZ ("Fark negatif")

ELSE

YAZ ("Fark pozitif")

END IF

BİTİR



h) Verilen iki sayının en büyük ortak bölenini bulan algoritmayı yazıp akış diyagramını çiziniz.

BAŞLA

OKU sayı1, sayı2

döngü = True

DO WHILE döngü = True

IF sayı1 > sayı2 THEN

temp = sayı1

sayı1 = sayı2

sayı2 = temp

ELSE IF sayı1 ≤ 0 THEN

sonuç = sayı2

döngü = False

ELSE

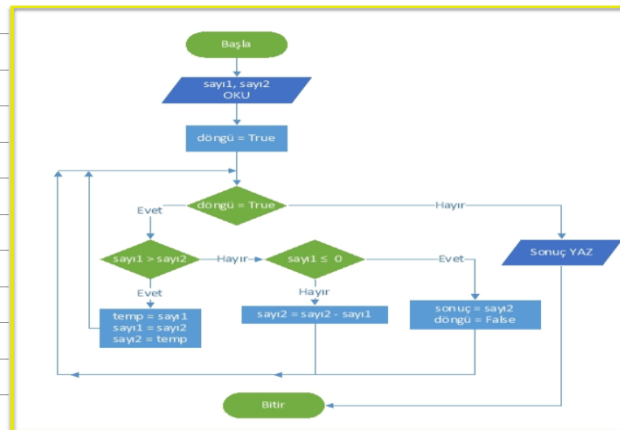
sayı2 = sayı2 - sayı1

END IF

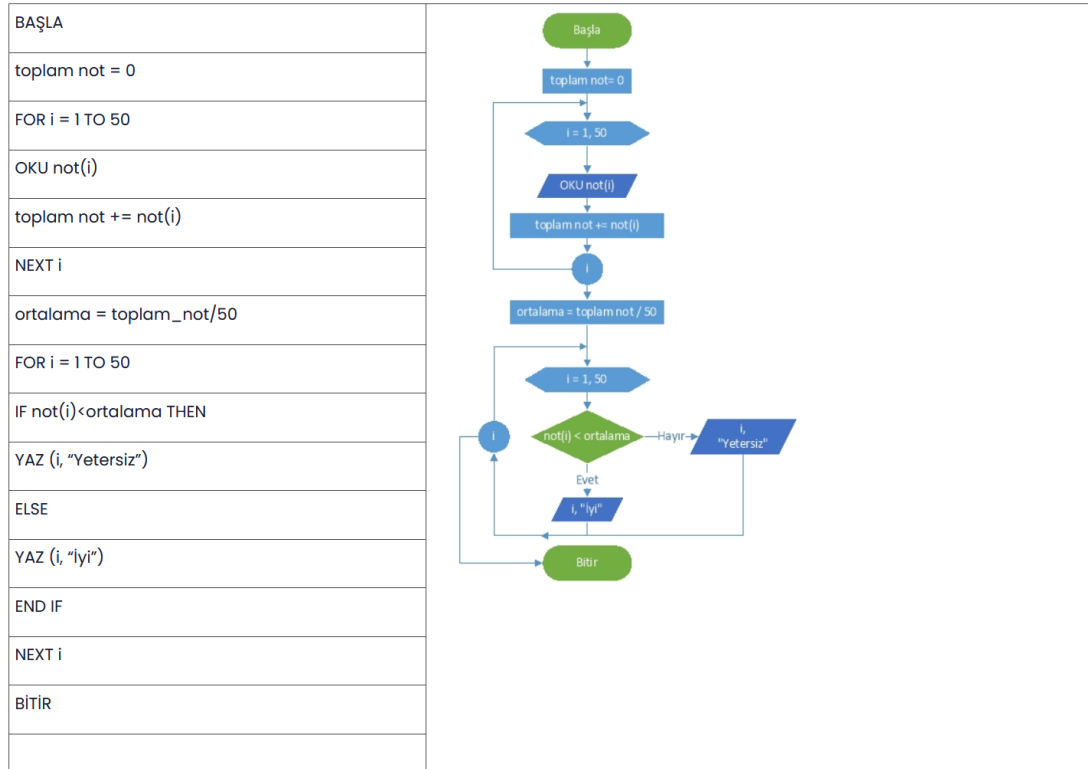
LOOP

YAZ sonuç

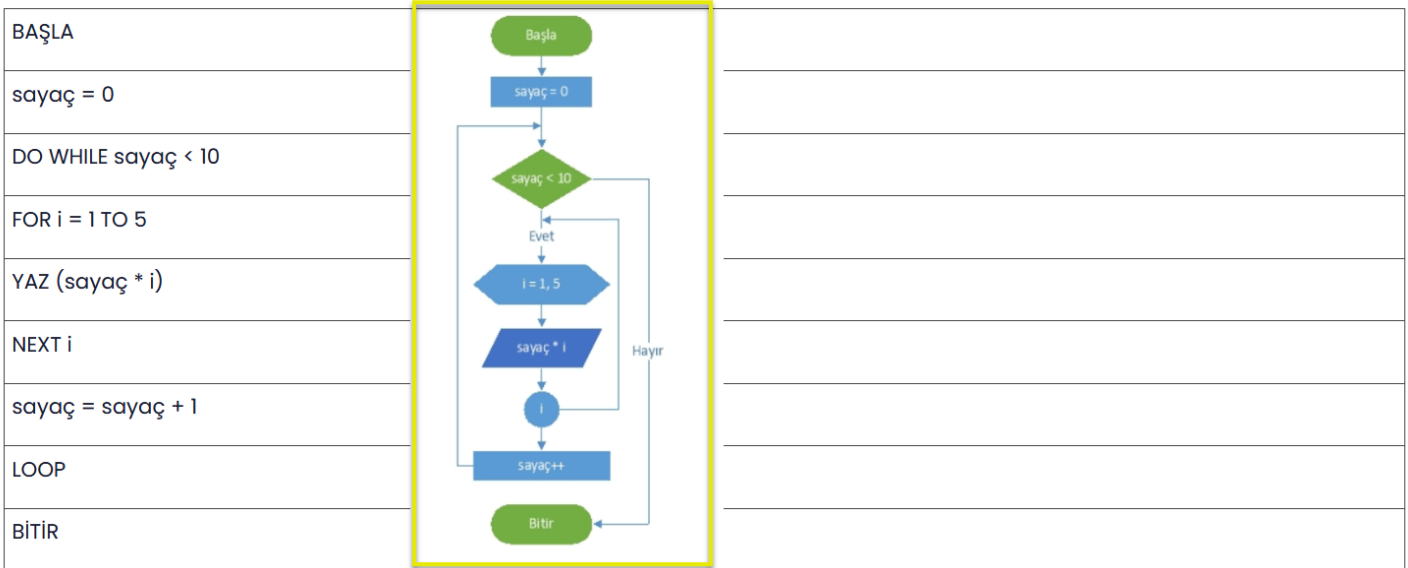
BİTİR



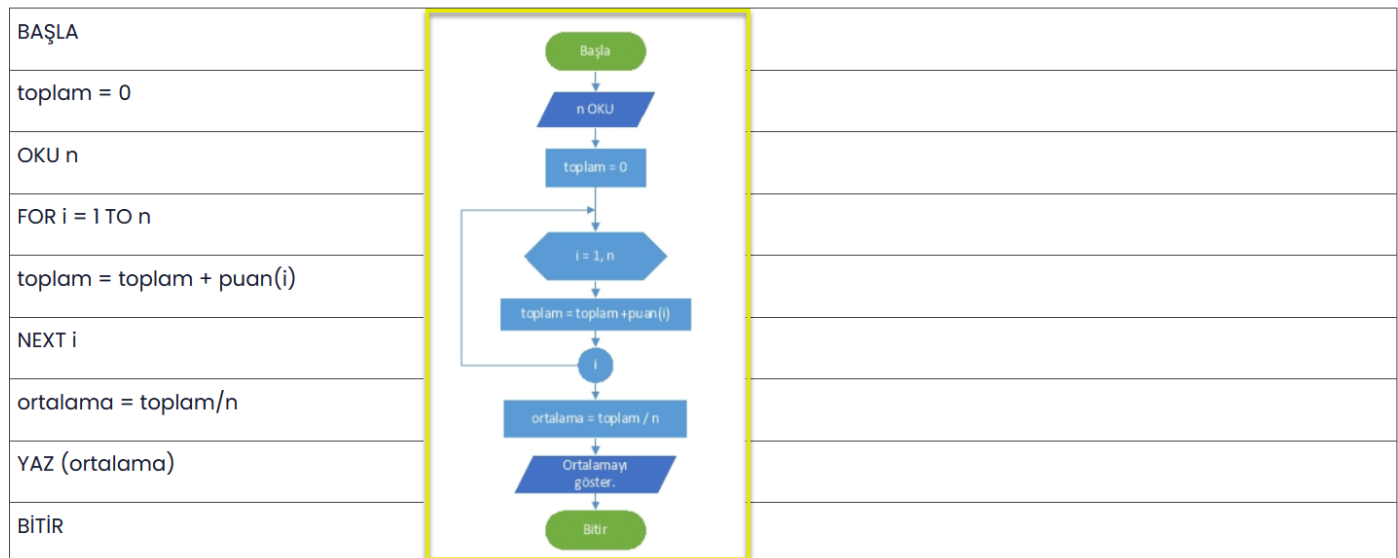
- i) 50 kişilik bir sınıfta yıl sonu notlarını kullanicidan olarak, sınıf ortalamasının altında kalanlar için "Yetersiz", üstünde veya eşit olanlar içinse "iyi" yazan algoritmayı ve akış diyagramını çiziniz.



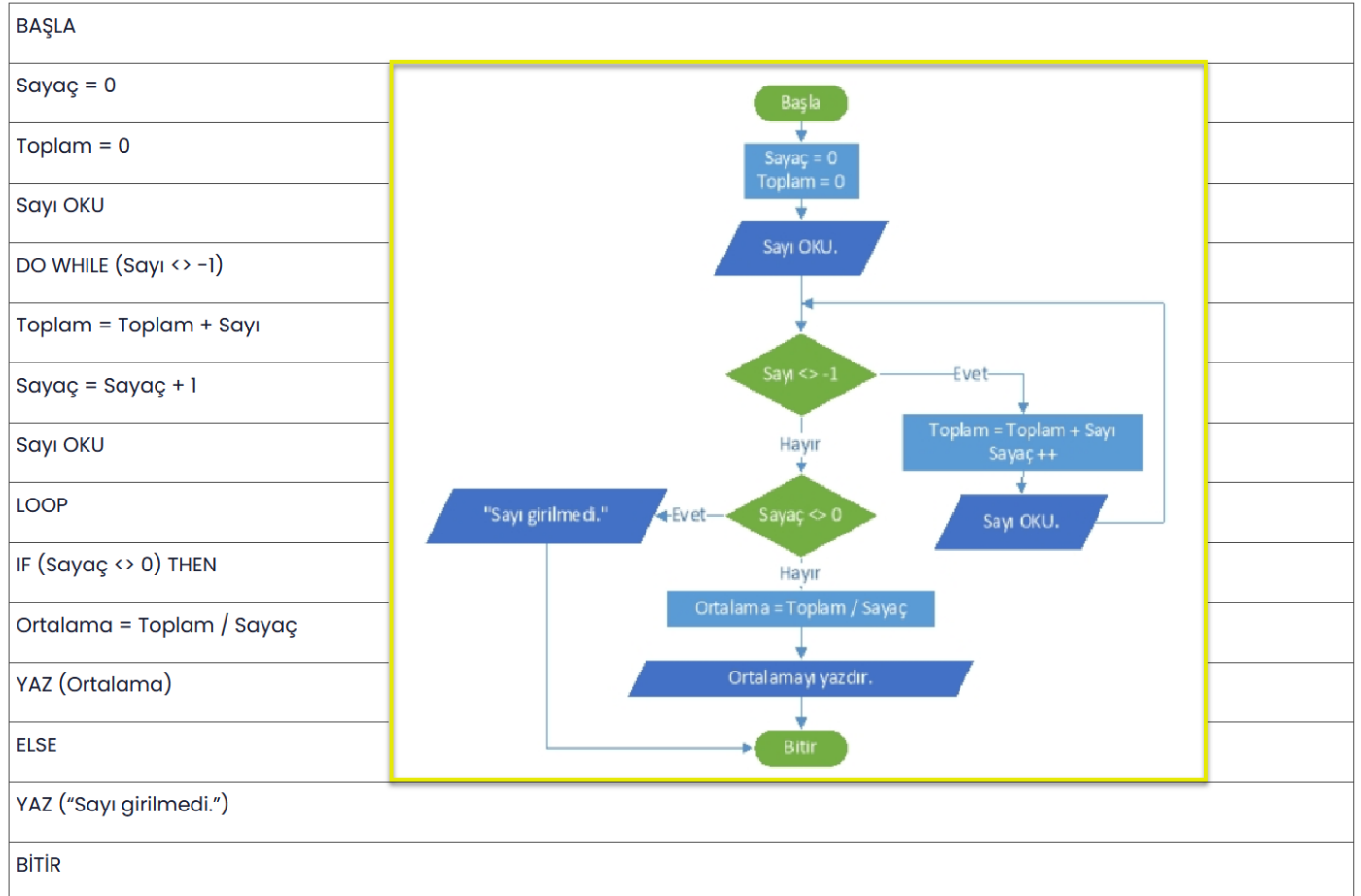
- j) 0'dan 10'a kadar olan her sayıyla 1'den 5'e kadar olan her sayının çarpımını ekrana yazdıran algoritma ve akış şemasını çiziniz.



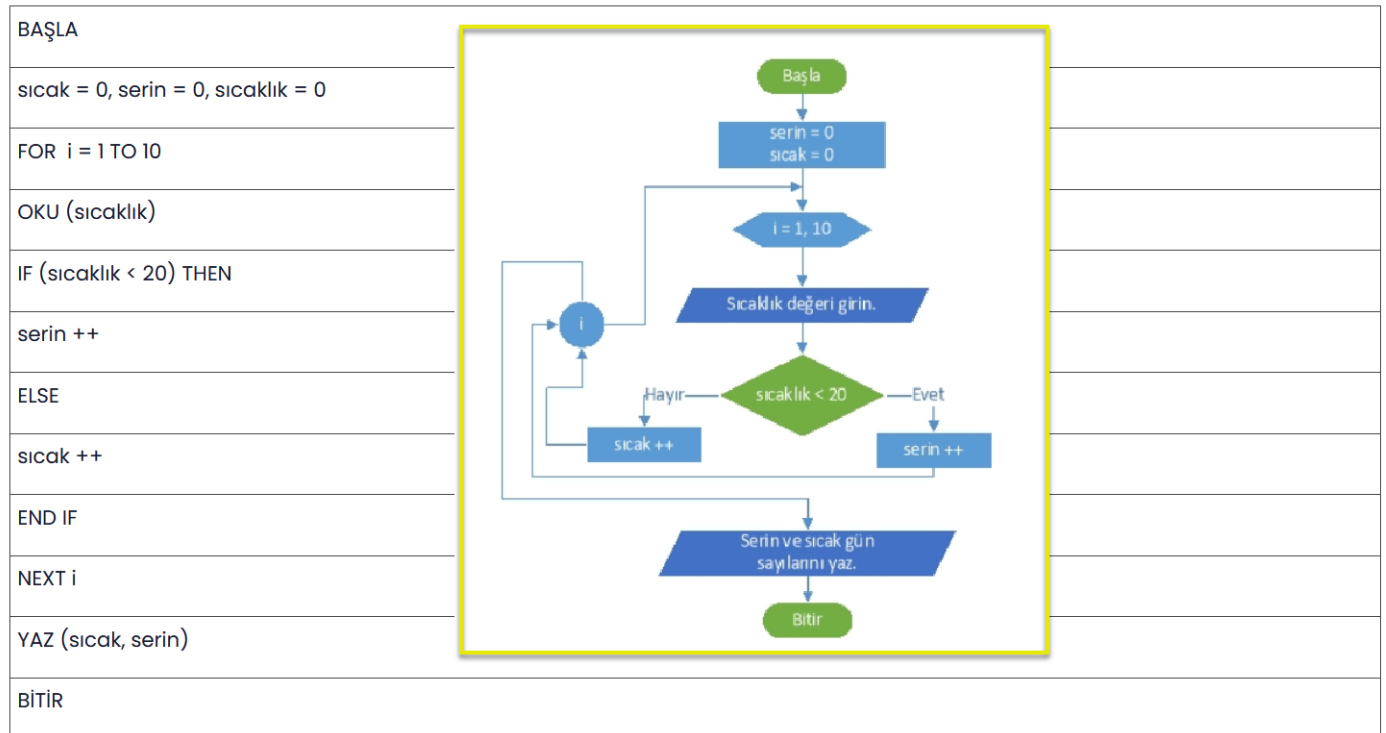
- k) Toplam n tane puan olan bir puan dizisinin ortalamasını bulan ve ekrana yazdıran algoritma ve akış şemasını çiziniz.



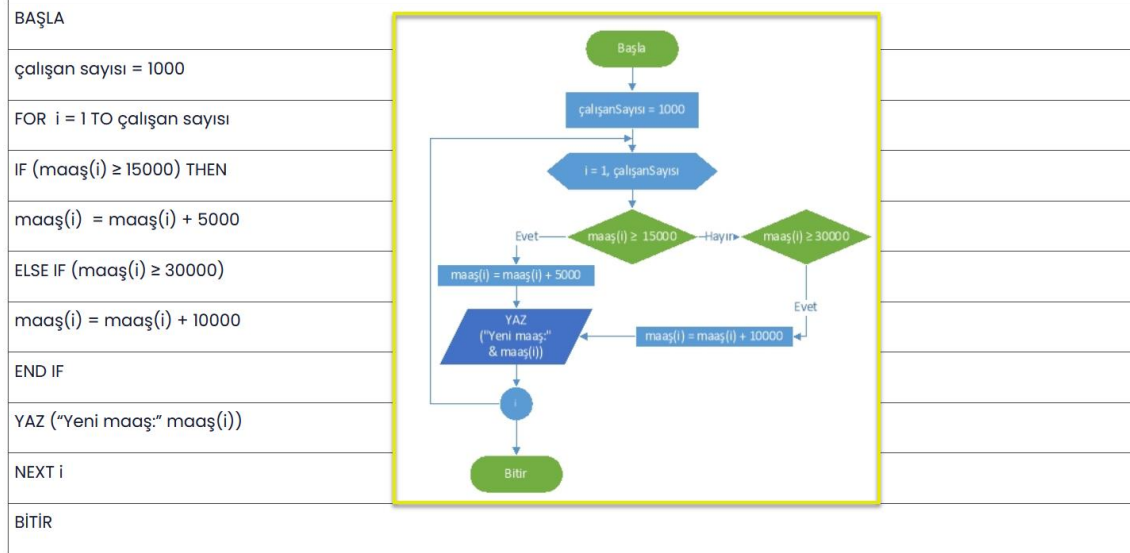
l) Kullanıcı tarafından girilen sayıların toplamını ve ortalamasını hesaplayan bir program aşağıda verilmiştir. Kullanıcı -1 değerini girmediği müddetçe, kullanıcıdan sayı girmesi istenir. -1 girildiğinde toplama işlemi sonlanır. Sayacın 0 olduğu durumda ise sayı girilmediği yazdırılır.



m) 10 güne ait hava sıcaklığı değerleri kullanıcı tarafından girilecektir. Bu değerlere bağlı olarak serin ve sıcak günlerin sayısını tutan programın kaba kodu ve akış diyagramı aşağıdadır.



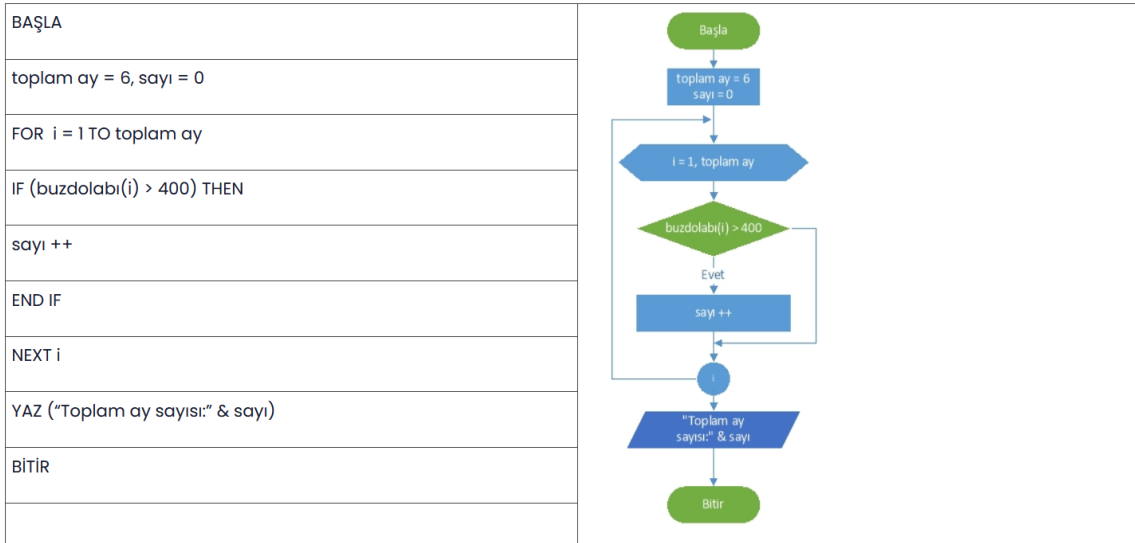
n) Bir şirketteki 1000 çalışanın maaşları bir dizi şeklinde verilmiştir. Çalışanlardan maaşları 15000 liraya eşit ve 15000 liradan yüksek olanlara 5000 lira, 30000 lira ve 30000 liradan yüksek olanlara 10000 lira zam yapılıp maaşların olduğu dizi güncellenecek ve en son tüm maaşlar yazdırılacaktır. Bunu yapan algoritma aşağıda verilmiştir.



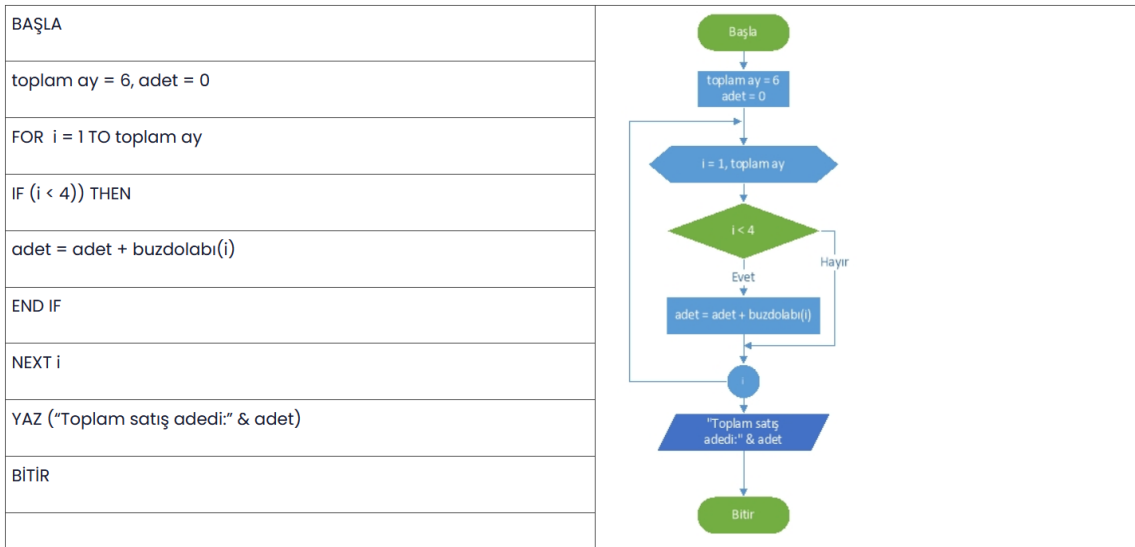
Ay	Ocak	Şubat	Mart	Nisan	Mayıs	Haziran
Buzdolabı adedi	350	560	360	450	400	900

o) Yukarıda bir firmanın ilk 6 ayındaki buzdolabı satış adetleri buzdolabı isimli dizide verilmiştir. Bu dizideki verilere göre üç farklı algoritma sorusu aşağıda verilmiştir.

- 400 adetten fazla buzdolabı satılan ayların toplam sayısı:



- İlk 3 aydaki toplam buzdolabı satış adedi:



- 300'den fazla 450'den az satış yapılan ayların sayısı:

BAŞLA	<pre> graph TD Start([Başla]) --> Init[toplam ay = 6 sayı = 0] Init --> SetI[i = 1, toplam ay] SetI --> Decision{buzdolabı(i) > 300 VE buzdolabı(i) < 450} Decision -- Evet --> IncSayı[sayı ++] Decision -- Hayır --> Loop IncSayı --> Loop Loop((i)) --> Output[/Ay sayısı: & sayı/] Output --> End([Bitir]) </pre>
toplam ay = 6, sayı = 0	
FOR i = 1 TO toplam ay	
IF (buzdolabı(i)>300 AND buzdolabı(i)<450) THEN	
sayı ++	
END IF	
NEXT i	
YAZ ("Ay sayısı:" & sayı)	
BİTİR	

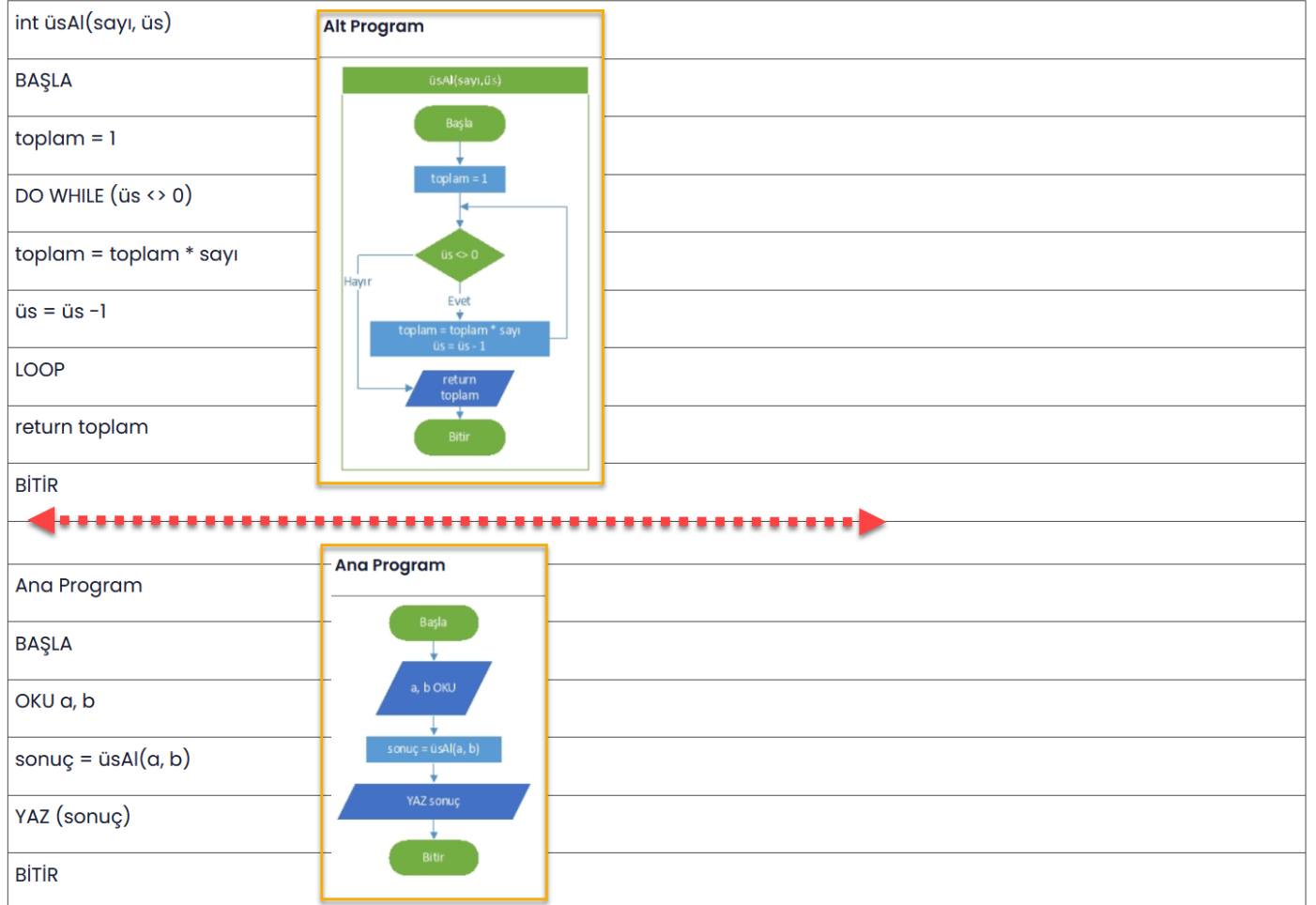
p) Fabrikadaki tüm ürünlerin maliyeti bir dizide verilmiştir. Bu fabrikada yeni üretilecek olan bir ürünün maliyetinin eski ürünlerin kaç tanesinin maliyetinden daha fazla olduğunu bulan algoritma ve akış diyagramı aşağıda verilmiştir.

BAŞLA	<pre> graph TD Start([Başla]) --> Read[/maliyet, ürün sayısı OKU/] Read --> Init[toplam sayı = 0] Init --> SetI[i = 0, ürün sayısı] SetI --> Decision{maliyet > dizi(i)} Decision -- Evet --> IncSayı[toplam sayı ++] Decision -- Hayır --> Loop IncSayı --> Loop Loop((i)) --> Output[/YAZ sayı/] Output --> End([Bitir]) </pre>
OKU maliyet, ürün sayısı	
toplam sayı = 0	
FOR i = 1 TO ürün sayısı	
IF (maliyet > dizi(i)) THEN	
toplam sayı ++	
END IF	
NEXT i	
YAZ (toplam sayı)	
BİTİR	

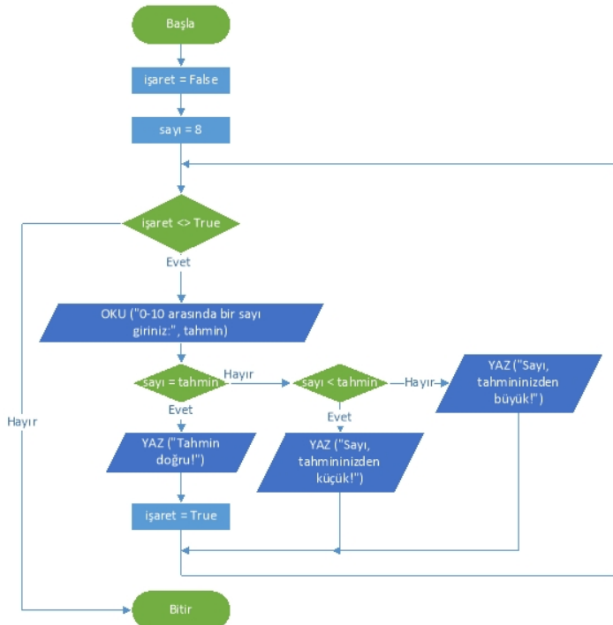
q) Girilen doğum yılına göre bir kişinin ehliyet alıp alamayacağını hesaplayan algoritma ve akış diyagramı aşağıda verilmiştir.



r) Girilen sayının girilen dereceden kuvvetini çarpma işlemiyle alan algoritma ve akış diyagramı aşağıda verilmiştir.



t) Kullanıcı tarafından tahmin edilen bir değer, sayının tanımlanmış değeri ile karşılaştırıldığı algoritma örneği ve akış diyagramı aşağıdadır.



s) Yarıçapı girilen kürenin yüzey alanını ve hacmini hesaplayan algoritma ve akış diyagramı aşağıda verilmiştir.



Not: Soru ve Cevaplar İstanbul Üniversitesi Öğrenme Yönetim Sistemindeki **Ünite Sonu Sorular** ve **Sorularla Öğrenelim** bölümünde kayıtlı olan en son hali ile kayıt olmuştur. Sonradan cevapların yönetim tarafından değiştirilmesi durumunda, derleyen kişi herhangi bir sorumluluk kabul etmez.

Her Öğrenci **VİZE** : Sınavı için 1-5 Üniteden Sorumludur **FİNAL**: Sınavı için 1-10 Üniteden Sorumludur.

Kaynak : İstanbul Üniversitesi AUEF -Ünite Sonu Soruları ve Sorularla Öğrenelim Soruları

TÜM ARKADAŞLARA BAŞARILAR DİLERİM