








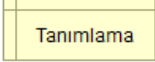
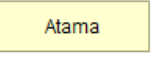
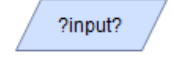
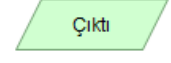

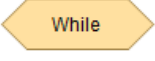
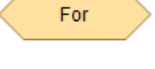
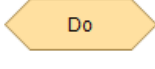
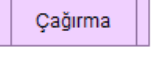
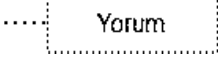

FLOWGORİTHM DERS NOTLARI:

Flowgorithm: Akış diyagramı şekilleri kullanarak basit kodlar yazma programıdır.

1. Program Arayüzü:

- a.  **Aç** : Önceden kayıtlı bir dosyayı yükler.
- b.  **Kaydet** : Mevcut çalışmayı kayıt eder.
- c.  **Çalıştır** : Akış diyagramını çalıştırır.
- d.  **Adım** : Kodları adım adım sırayla çalıştırır. (BP)
- e.  **Fonksiyon Yöneticisi** : Ana program veya alt programlar arası geçişi sağlar.
- f.   **Büyüteç** : Sayfayı büyütür ya da küçültür.

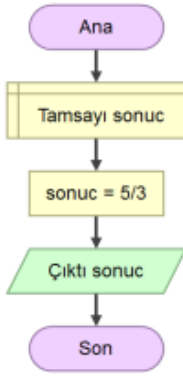
2. Akış Diyagramı Şekilleri :

- a.  : Kullanılacak değişkenleri sisteme söyler.
- b.  : Tanımlanmış değişkene değer atar.
- c.  : Klavyeden bilgi girişi için kullanılır.
- d.  : Ekranı bilgi yazdırmak için kullanılır.
- e.  : Karar verme işlemleri için kullanılır.
- f.  : Tekrar sayısı belli olmayan DÖNGÜ için kullanılır.
- g.  : Tekrar sayısı belli olan DÖNGÜ için kullanılır.
- h.  : En az 1 Kere çalışması gereken DÖNGÜ için
- i.  : Tanımlı altprogramı çağırma için kullanılır.
- j.  : Kodların çalışmasını ETKİLEMEYEN, programcı için AÇIKLAYICI BİLGİLER yazmak için kullanılır.
- k.  (KIRILMA NOKTASI / BREAK POINT): **Hata ayıklama** işlemlerinde çalışan programın bu noktaya geldiğinde DURAKLAMASINI sağlar. Programın çalışması istenildiğinde Normal olarak (▶) ya da (ADIM ADIM ▶) devam ettirilir.

3. Veri Tipleri:

a. **Integer** : Tam sayı verileri saklamak için kullanılır.

-10 3 100 10000000000 (taşma)



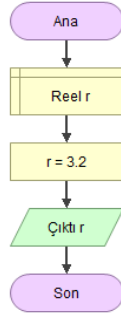
(Çıktı neden 1 olur?)

b. Real (Reel): Ondalıklı sayıları saklamak için kullanılır.

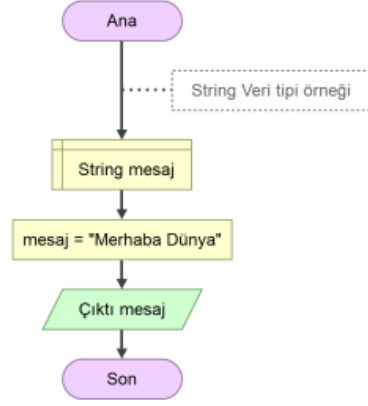
~~3,2~~ Virgöl OLMAZ

3.2 Doğru

$5/5 = 1.0$ (bölme işlemi sonucu her zaman reel)



c. **String** : Herhangi text veriyi saklamak için kullanılır.
“Hoşgeldiniz” “1. Sayıyı giriniz:”



d. **Boolean** : Sadece True/False ya da 0/1 değerini saklar.

(İF ile birlikte iyi gider 😊)

sonuc = 5>3 (sonuc TRUE olur)

önce eşitliğin sağ tarafı

sonuc = 1>2 (sonuc FALSE olur)

4. Operatörler:

a. Matematiksel: İşlem Önceliğine Dikkat

- i. Toplama + : 3+2
- ii. Çıkarma - : 9-7
- iii. Çarpma * : 6*3
- iv. Bölme / : 9/3
- v. Mod % : 10%3
- vi. Üs Alma ^ : 4^3

Level	Name	Operators	Notes
8	Unary	- ! not	In Visual Basic, "not" precedence level is far lower - above "and", but below all relational operators.
7	Exponent	^	The exponent operator does not exist in C# or Java.
6	Multiply	* / % mod	Division will always be high-precision (floating point)
5	Addition	+ -	In Flowgorithm, "+" will only work with numbers.
4	Concatenate	&	C# and Java use the ambiguous "+" operator for addition and concatenation.
3	Relational	> >= < <= == = != <>	
2	Logical And	and &&	
1	Logical Or	or	

b. Mantıksal Operatör

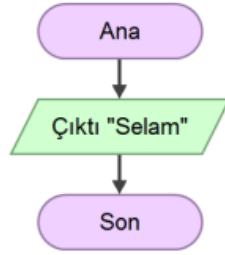
- i. Eşit Mi == 3==3 > True
- ii. Eşit Değil mi != <> 4!=5 True
- iii. Büyük mü >
- iv. Büyük Eşit mi >=
- v. Küçük mü <
- vi. Küçük Eşit mi <=
- vii. Veya || OR
- viii. Ve && AND
- ix. Değil ! NOT

c. String Birleştirme (Concatenation) &

"Merhaba" & "Dünya"

Merhaba String ifadesi ile Dünya String ifadesini birleştirir. MerhabaDünya String ifadesini oluşturur.

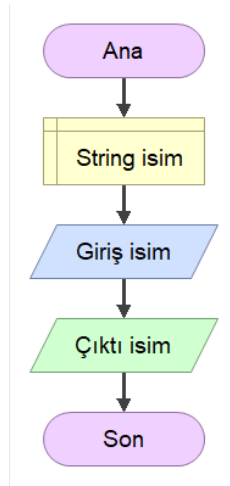
Örnek-1: Ekranaya Selam Yazan program.



Örnek-2: Kullanıcı klavyeden adını girecek.

Program da "Merhaba X" olarak ekranaya çıktı üretecek.

(x kişinin girdiği adını temsil ediyor. Herkes farklı isim girebilir.)



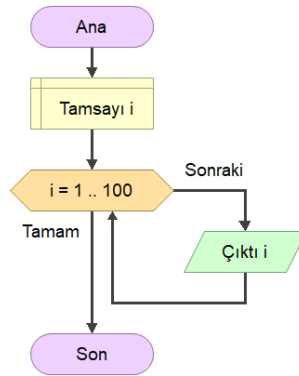
Açıklama [WK1]: Sol taraftaki algoritmayı, Merhaba XXXXXX şeklinde ekranaya çıktı verecek şekilde düzeltiniz.

DÖNGÜ: Tekrar eden yapılardır.

Tekrar Sayısı Belli	Tekrar Sayısı Belli Değil
FOR	WHİLE

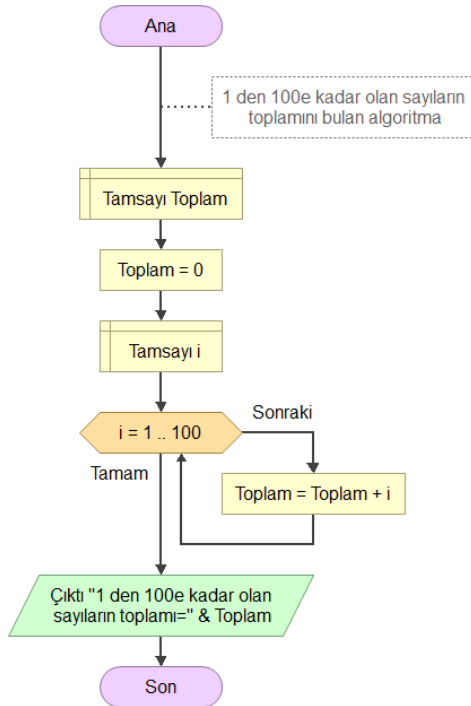
Örnek-3: 1 den 100 e kadar (100 dahil) tam sayıları ekrana yazdırın. Tekrar sayısı belli bir döngü.

Açıklama [WK2]: Başlangıcı=1
Bitiş=100
Adım=1
Tekrar sayısı belirli olduğu için
FOR döngüsü



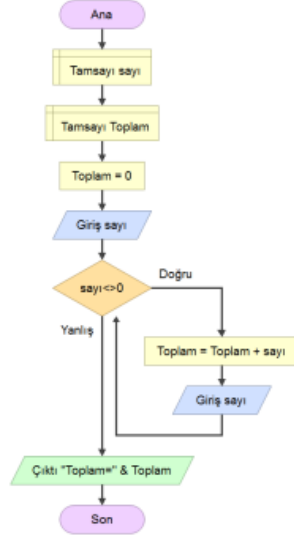
Örnek-4: 1 den 100 e kadar olan sayıların **TOPLAMI.** (Tekrar sayısı belli)

Açıklama [WK3]: Toplam = Toplam + i
teknîği ile, her i değeri, Toplam değişkenine
ilave edilir.



Örnek-5: Klavyeden 0 girilene kadar girilen sayıları TOPLAYAN program.

Açıklama [WK4]: Kullanıcının 0'ı ne zaman gireceği belli değil. Belki ilk seferde, belki 10 sayıdan sonra. Onun için tekrar sayısı belli değil
WHILE DÖNGÜSÜ



Örnek-6: Klavyeden Girilen 3 Ayrı Sayının Bir Üçgen Oluşturup Oluşturmayacağını Tahmin Eden Programı Yazınız.

$$(a,b,c) \quad |a-b| < c < |a+b|$$

HAZIR FONKSİYONLAR:

ABS(n) : Sayısal ifadenin Mutlak değeri bulur. $Abs(-5) = 5$

LEN(s) : String ifadenin karakter sayısını verir. $Len("Selam") = 5$

Char(s, i) : String verisindeki, i. Karakteri verir.

(0. Karakter ilk karakterdir)

$Char("Selam", 3) \rightarrow a$

$Char("Selam", 1) \rightarrow e$

$Char("Merhaba", 5) \rightarrow b$

Random(n) : 0 .. (n-1) aralığında rastgele TAMSAYI sayı üretir.

- $Random(10)$ hangi aralıkta rastgele tamsayı üretir?
- $Random(29)$ hangi aralıkta rastgele tamsayı üretir?
- $Random(100) + 5$ hangi aralıkta rastgele tamsayı üretir?
- 0-99 aralığında rastgele tamsayı üreten komut?
- 10-100 aralığında rastgele tamsayı üreten komut?
- $Random(10) + Random(10) \rightarrow Random(19)$
- 0 dan 20 ye kadar rastgele $\rightarrow Random(21)$

ToChar(n) : Girilen sayının ASCII karakter tablosundaki karşılığını verir.

- $ToChar(65) \rightarrow A$

- $ToChar(97) \rightarrow a$

ToCode(c) : 1 Karakterin, sayısal kod karşılığını verir.

- $ToCode("A") \rightarrow 65$

- $ToCode("L") \rightarrow 76$

- $ToCode("FEN") \rightarrow ???$ Neden çalışmaz

Size(array) : Bir dizinin, eleman sayısını verir. DİZİLER konusunda işlenecek.

VERİ TİPİ DÖNÜŞTÜREN HAZIR FONKSİYONLAR:

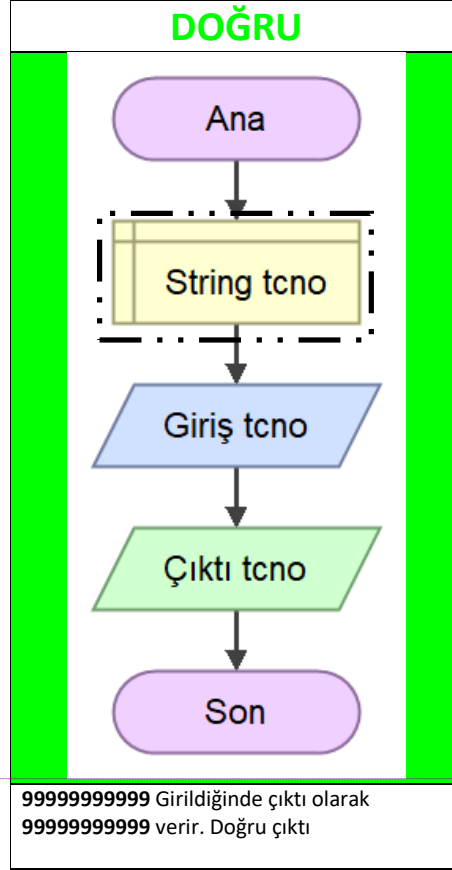
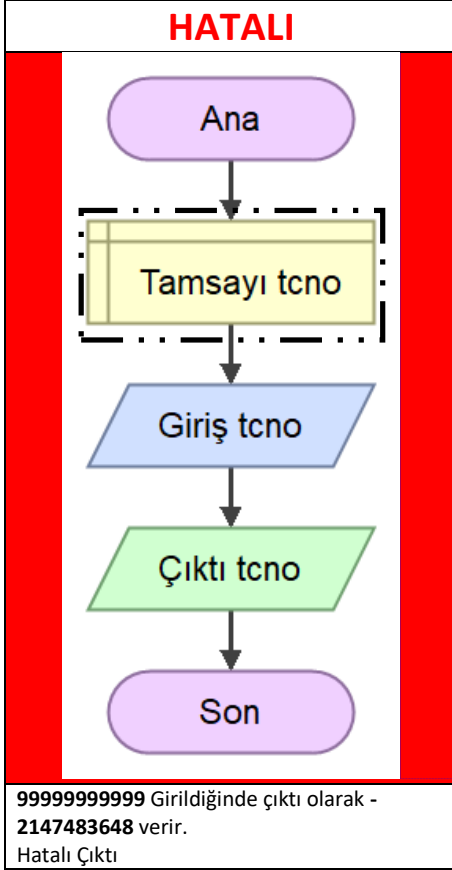
ToInteger(s) : String → Integer' a dönüştürür.
ToReal(s) : String → Real' a dönüştürür.
ToString(n) : Integer → String' e dönüştürür.

Örnek: Klavyeden girilen T.C. Kimlik numarasının

-11 karakter uzunluğunda olduğunu (LEN fonksiyonu ile çok kolay kontrol edilir.)

-son rakamının ÇİFT SAYI olduğunu kontrol eden programı yazınız.

(TC Kimlik No verisini neden STRING ifade olarak saklamalıyız?)



Açıklama [WKS]: 2 algoritma arasındaki tek fark tcno değişken tipinin birisinde String birisinde Tamsayı olarak tanımlanmış olmasıdır.

Peki STRING olarak girilen tcno verisi doğru teknik ise soruda istenen "son rakamın çift olma" zorunluluğunu nasıl kontrol edersin?

1-Son karakteri Char() hazır fonksiyonu ile elde ederiz.
sonKr = Char(tcno,11)

2-Elde edilen değer STRING bir ifadedir. Çift olup olmadığını bulmak için bu değer 2'ye göre MOD' unu almak gerekir. Ama MOD() fonksiyonu sayısal veriler üzerinde çalışır. O zaman, 1. Adımda elde ettiğim String değeri TAMSAYIYA yani INTEGER değere dönüştürmem gerekir.
sonKr = ToInteger(Char(tcno,11))

3-Artık 2 ye göre MOD' unu alabilirim.
SonKr = ToInteger(Char(tcno,11)) % 2

4- Eğer sonKr==0 ise son rakam ÇİFTTİR kurala uygundur, değilse TEKİR ve kurala uygun değildir.

DİZİLER (Arrays):

Açıklama [06]: Programlamada ARRAY olarak geçer. LÜTFEN ARAŞTIRINIZ.

ISINMA SORUSU: 1 öğrencinin 10 tane dersi varsa, her dersten 2 yazılı, 2 sözlü , 1 proje ve Not Ortalaması (Her bir ders için) kaç değişken tanımlamak gerekir?

1 Öğrenci için -> 10 Ders x (2+2+1+1) = 60 tane değişken

1 Sınıf için (30 Öğrenci) -> 60 x 30 = 1800 tane değişken

1 Okul için (12 Şube) -> 1800 x 12 = **21.600**

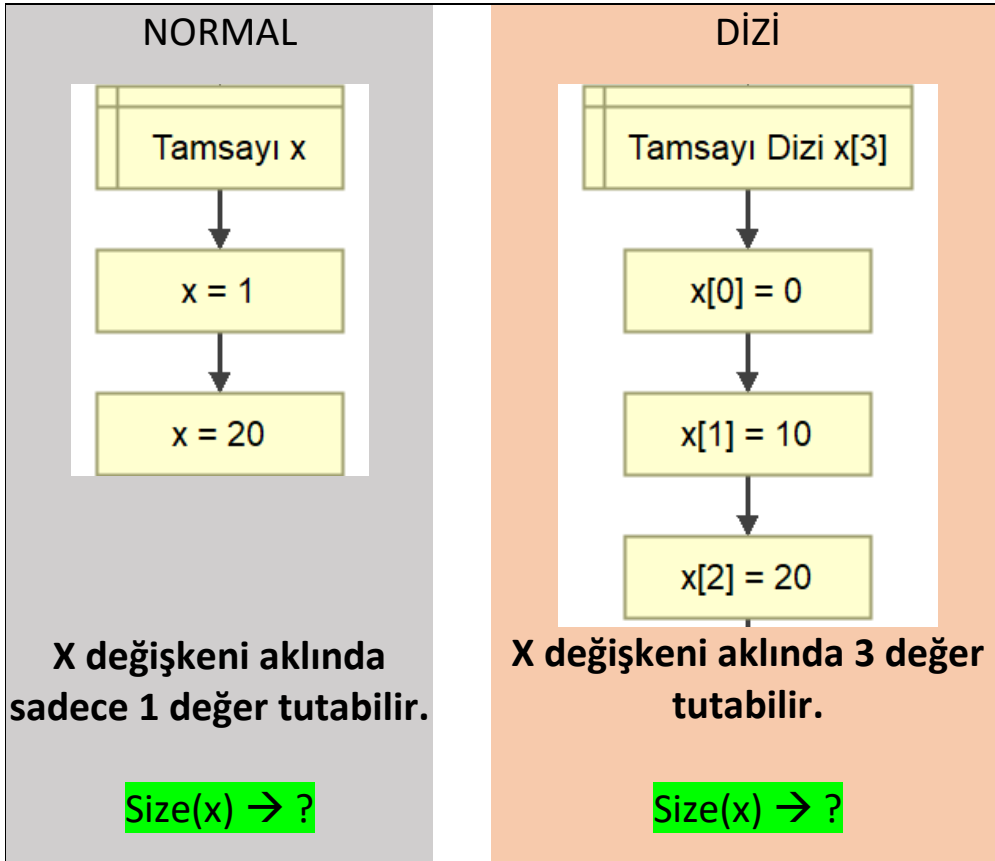
OMG Bunun kolay bir yolu olmalı

Dizi: Aynı tipteki verileri tek bir deęişkende saklamak için kullanılan yapılardır.

(Aşağıda yazan bilgiler Flowgorithm programı için geçerlidir, başka programlama ortamları için deęişebilir).

- Dizi boyutu önceden belirtilmelidir.
- Dizi elemanlarına erişmek için [] işaretleri kullanılır.
- Dizinin ilk elemanı **0. elemandır.**
- DÖNGÜ ile birlikte çok iyi gider.

Size(array) : Kendisine gönderilen dizinin eleman sayısını verir. (Hazır fonksiyondur)

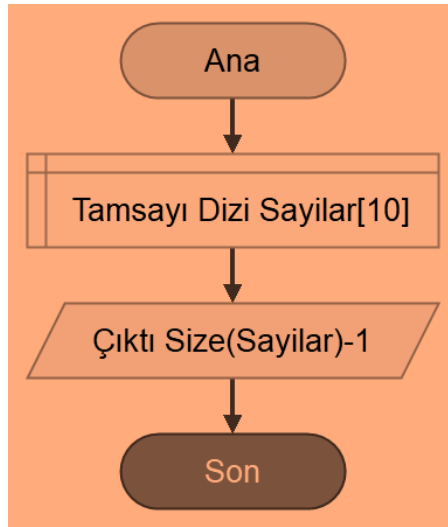
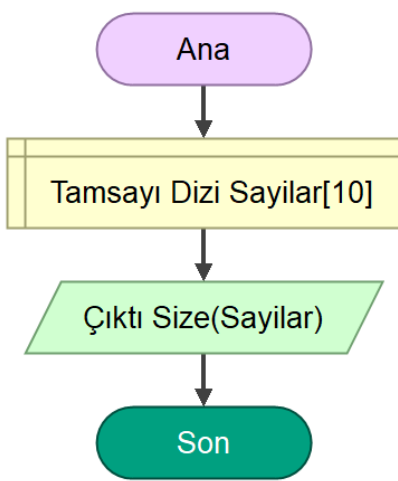


Çalışma Örnekleri:

- 3 elemanlı **Integer** tipte, **SAYILAR** isimli diziyi oluşturun.
- String** tipte, **OGRENCI** isimli, **5** elemanlı diziyi oluşturun.
- 100** elemanlı **Real** tipte, **NOTLAR** isimli diziyi oluşturun.
- SONUC** isminde, **Boolean** tipte, **2** elemanlı diziyi oluşturun.

Açıklama [WK7]: Aşağıdaki tanımlara uygun değişken tanımlamalarını "FLOWGORITHM Tanımlama" bloğu ile gösteriniz.

Örnek: Aşağıdaki program çıktısı ne olur?



Örnek-1: 10 elemanlı, Integer tipte, DEGERLER isimli bir dizi oluşturun. Bu dizinin içine rastgele 0..100 aralığında sayılar ekleyin.

- ➔ Bu dizinin TOPLAMINI / ORTALAMASINI / MIN..MAK değerlerini bulunuz.
- ➔ $Toplam = Toplam + DEGERLER[i]$

Örnek-2 ORTA: 8 elemanlı, String tipte SESLIHARFLER isimli bir dizi oluşturun. a,e,i,i,o,ö,u,ü harflerini tek tek bu diziye yükleyin.

Örnek-3 ZOR: Klavyeden Girilen isim verisindeki tüm karakterleri HARFLER isimli diziye aktaran algoritmayı yazınız.

ÖMER	HARFLER[?]	
	0	Ö
	1	M
	2	E
	3	R

→ Örnek 2 deki algoritmayı kullanarak, örnek 3 te girilen isimdeki sesli karakteri ekrana yazınız.

Soru: 5 elemanlı A dizisinin son elemanını 3 yapan kodu yazınız.

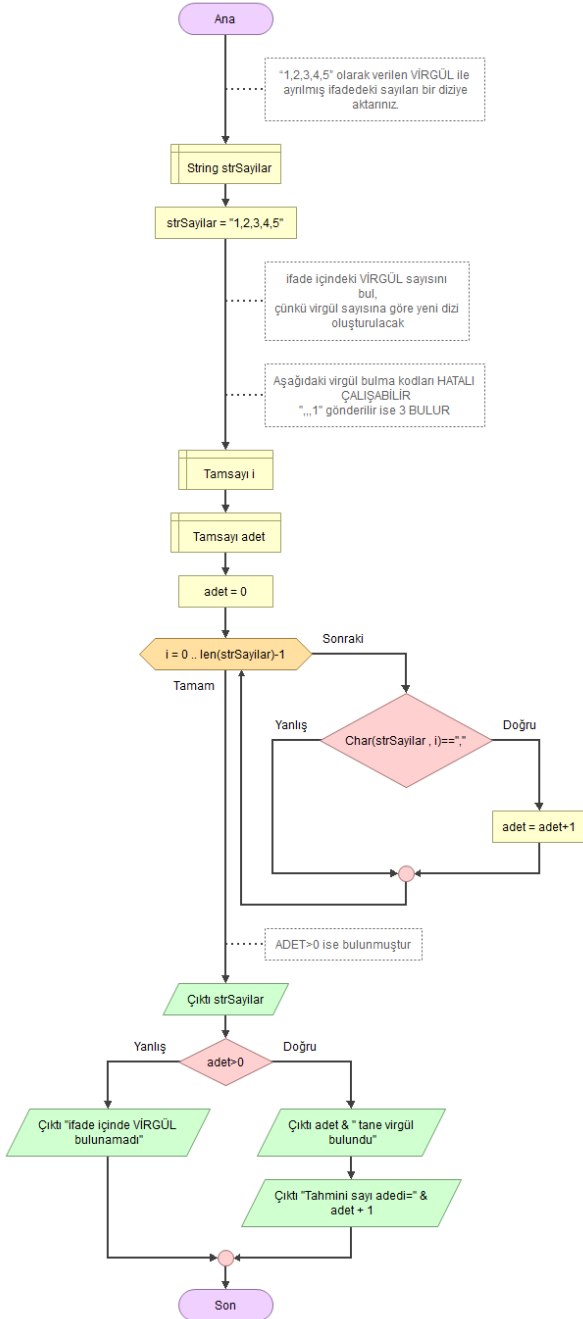
Soru: eleman sayısı bilinmeyen OGR dizinin SON ELEMANINA nasıl erişirsin?

Soru: $Toplam = Toplam - K[3]$ komutu ne işe yarar açıklayınız?

Toplam değişkenini, K dizisinin 4. Elemanı kadar azaltır.

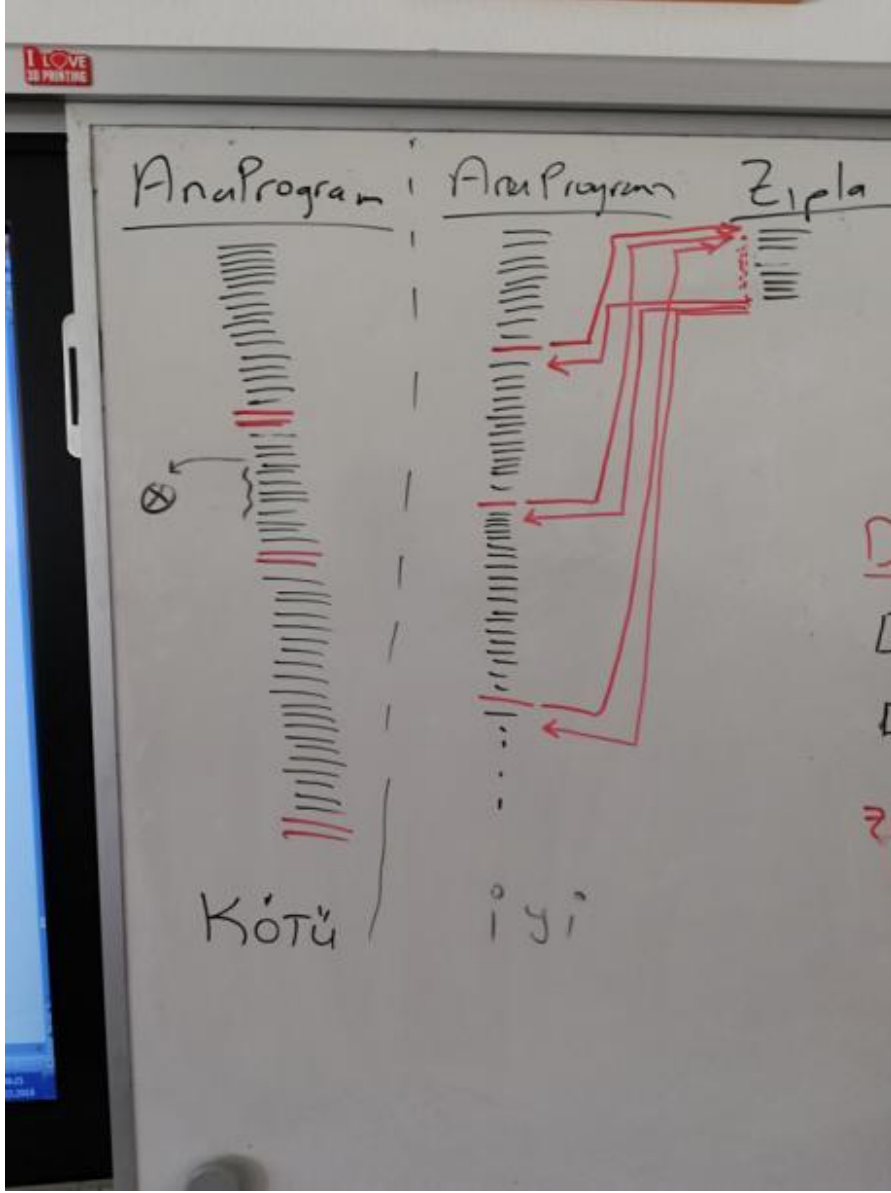
Soru ORTA: Klavyeden girilen isim verisindeki her karakterin ASCII kod karşılığının toplamını bulan programı yazınız?

SORU ZOR: "1,2,3,4,5" olarak verilen VİRGÜL ile ayrılmış ifadedeki sayıları bir diziye aktarınız.




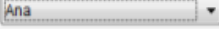
KULLANICI TANIMLI FONKSİYONLAR (Alt Programlar):

Belli bir görevi gerçekleştirmek için yazılmış kodlardır.



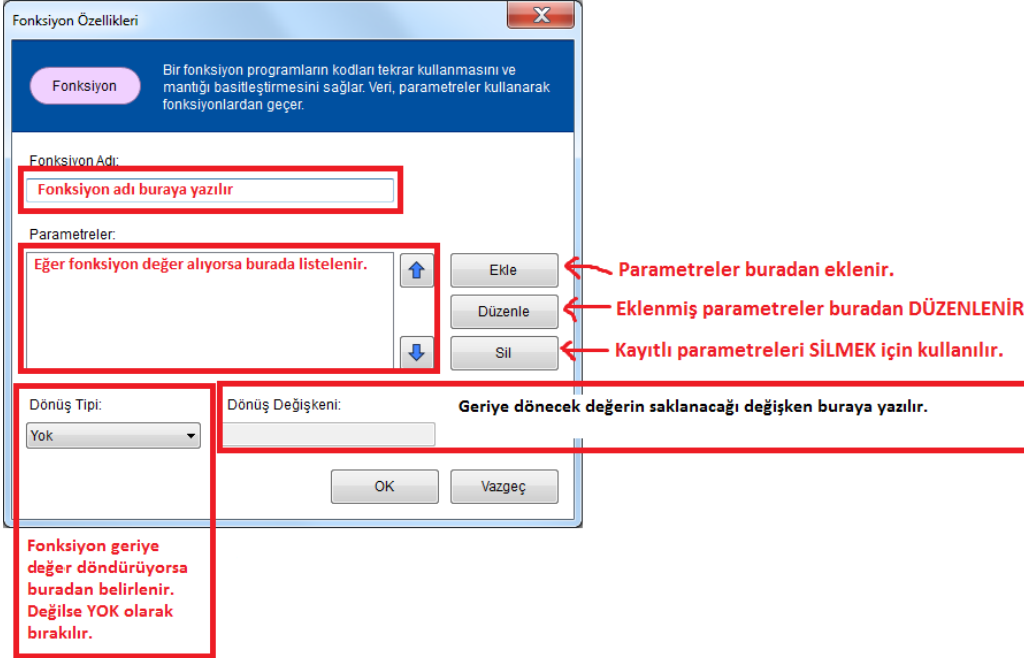
“Eğer belirli bir kod bloğunu sürekli tekrar tekrar yazıyorsan orada bir hata vardır” Ö. ERMiş 😎

 **Fonksiyon Yöneticisi:** Fonksiyon oluşturmak, düzenlemek veya silmek için kullanılır.

 **Fonksiyon Listesi:** Ana program ile fonksiyonlar arasında geçiş için kullanılır.

 **Çağırma:** Tanımlı altprogramı çağırma için kullanılır.

GeriyedeĞER DÖNDÜRMEYEN fonksiyonlar çağrılabilir.



Fonksiyon Özellikleri

Fonksiyon Bir fonksiyon programların kodları tekrar kullanmasını ve mantığı basitleştirmesini sağlar. Veri, parametreler kullanarak fonksiyonlardan geçer.

Fonksiyon Adı:

Parametreler:

Dönüş Tipi:

Dönüş Değişkeni:

Fonksiyon geriye değer döndürüyorsa buradan belirlenir. Değilse YOK olarak bırakılır.

Parametreler buradan eklenir.
Eklenmiş parametreler buradan DÜZENLENİR.
Kayıtlı parametreleri SİLMEK için kullanılır.

GİRDİ DURUMUNA GÖRE:

Değer ALMAYAN fonksiyonlar:

Örnek: Goster() Zıpla()

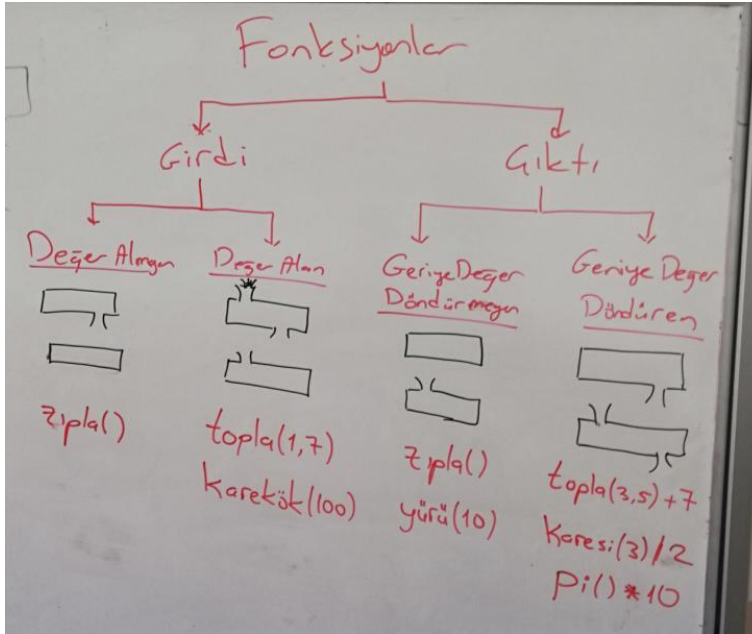
1 veya daha fazla değer alan fonksiyonlar:

Örnek: KareAl(5) Topla(3,5)

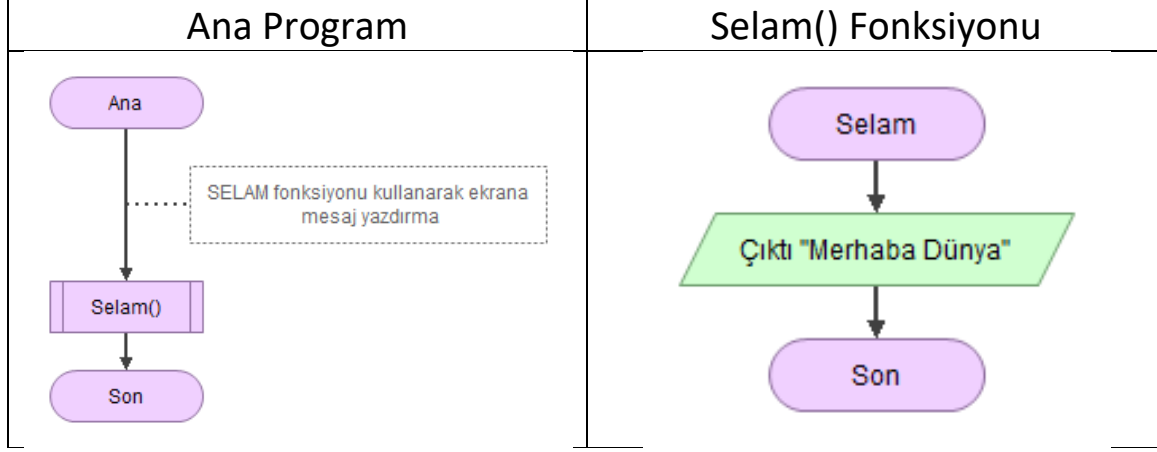
ÇIKTI DURUMUNA GÖRE (Geri değer döndürme durumu):

Değer DÖNDÜRMEYEN fonksiyonlar:

1 veya daha fazla değer alan fonksiyonlar:



Örnek-1 : Ekranı "Merhaba Dünya" yazan Selam() fonksiyonunu oluřturunuz.



Fonksiyon Penceresi Görünümü:

Fonksiyon Özellikleri

Fonksiyon Bir fonksiyon programların kodları tekrar kullanmasını ve mantığı basitleřtirmesini saęlar. Veri, parametreler kullanarak fonksiyonlardan geęer.

Fonksiyon Adı:
Selam

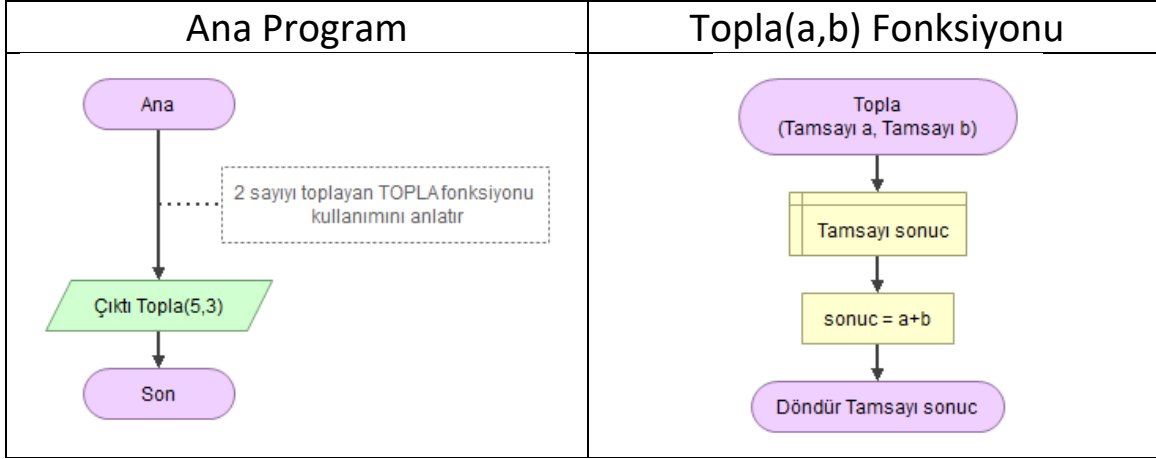
Parametreler:

↑ Ekle
Düzenle
↓ Sil

Dönüş Tipi: Yok
Dönüş Deęişkeni:

OK Vazgeç

Örnek-2: Kendisine gönderilen tamsayı tipte 2 veriyi toplayıp, geriye sayıların toplamını döndüren Topla(a , b) fonksiyonunu oluşturunuz.



Fonksiyon Penceresi Görünümü:

Fonksiyon Özellikleri

Fonksiyon Bir fonksiyon programların kodları tekrar kullanmasını ve mantığı basitleştirmesini sağlar. Veri, parametreler kullanarak fonksiyonlardan geçer.

Fonksiyon Adı:

Parametreler:

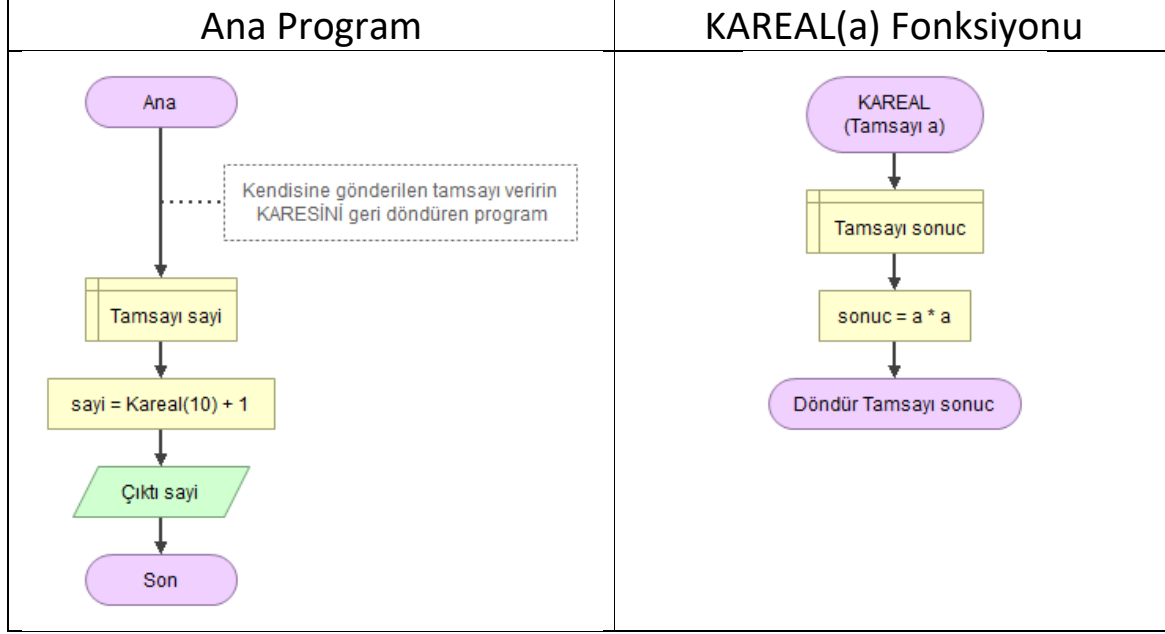
Tamsayı a	↑	Ekle
Tamsayı b		Düzenle
	↓	Sil

Dönüş Tipi:

Dönüş Değişkeni:

OK Vazgeç

Örnek-3: Kendisine gönderilen tamsayı tipte sayının KARESİNİ hesaplayıp geri döndüren KAREAL(a) fonksiyonunu oluşturunuz.



Fonksiyon Penceresi Görünümü:

Fonksiyon Özellikleri

Fonksiyon Bir fonksiyon programların kodları tekrar kullanmasını ve mantığı basitleştirmesini sağlar. Veri, parametreler kullanarak fonksiyonlardan geçer.

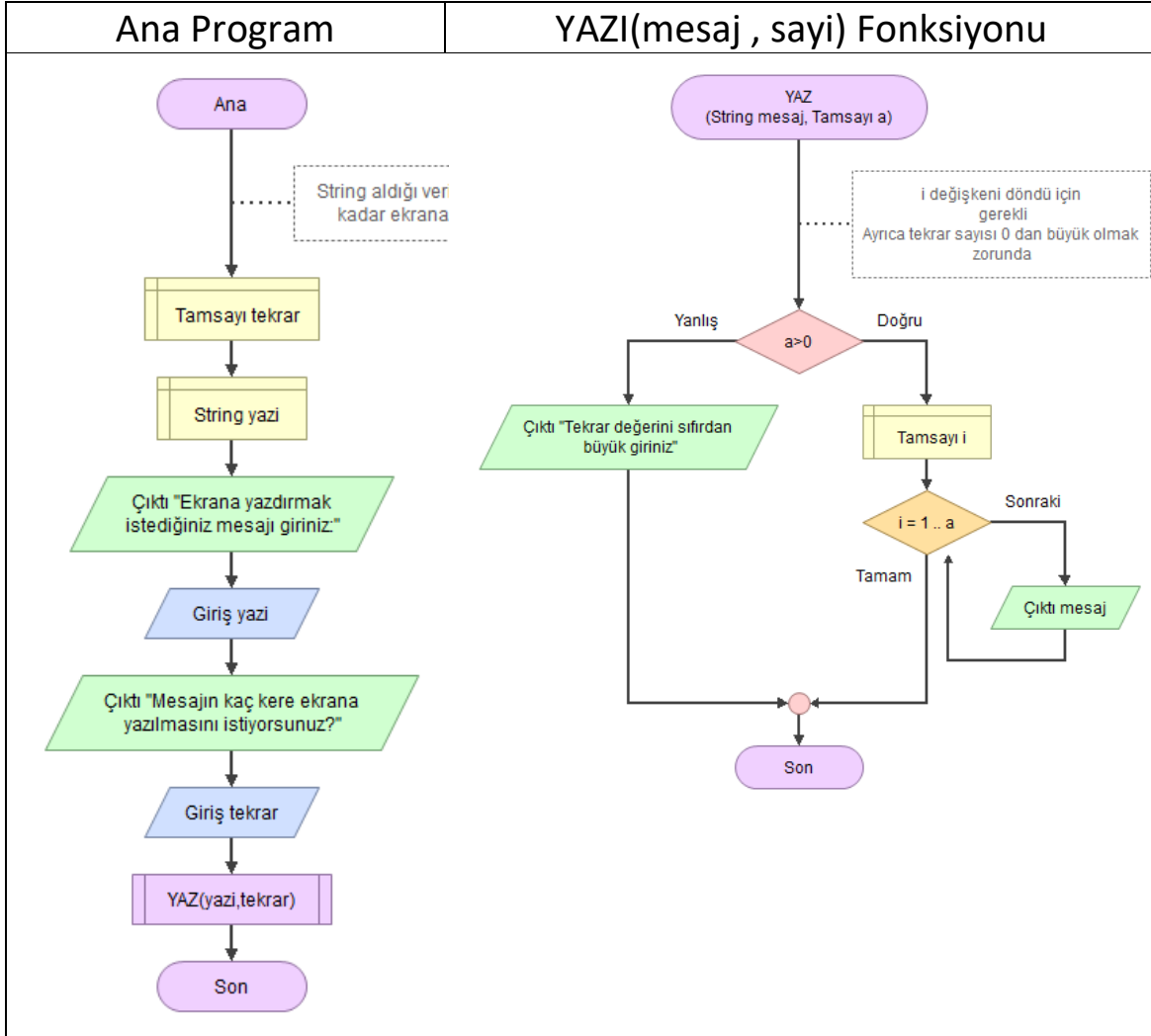
Fonksiyon Adı:

Parametreler:

Tamsayı a

Dönüş Tipi: Dönüş Değişkeni:

Örnek-4: String tipteki mesajı, Ekranı belirlenen sayı kadar yazdıran YAZ(mesaj , sayı) fonksiyonunu oluşturunuz.



~~ÖRNEK ->: String tipteki mesajın uzunluğunu ekrana yazan UZUNLUK fonksiyonunu tasarlayınız? (LEN fonksiyonu kullanarak)~~

SORU:-> Kendisine gönderilen 2 tane String ifadenin uzunluklarının çarpımını veren fonksiyon tasarlayınız. (UZUNLUKCARP)

Fonksiyon Penceresi Görünümü (örnek-4):

Fonksiyon Özellikleri

Fonksiyon Bir fonksiyon programların kodları tekrar kullanmasını ve mantığı basitleştirmesini sağlar. Veri, parametreler kullanarak fonksiyonlardan geçer.

Fonksiyon Adı:
VAZ

Parametreler:
String mesaj
Tamsayı a

Ekle
Düzenle
Sil

Dönüş Tipi: Yok
Dönüş Değişkeni:

OK Vazgeç

SORU: ekrana 100 defa SELAM yazan programı ALT program (fonksiyon) kullanarak tasarlayınız.

SORU: Kendisine gönderilen Tamsayı tipteki DİZİ içindeki elemanların SAYISINI (Dizide kaç tane eleman var) geriye döndüren fonksiyon yazınız.

SORU: Kendisine gönderilen Tamsayı tipteki DİZİ içindeki elemanların TOPLAMINI geriye döndüren fonksiyon yazınız.

SORU: Kendisine gönderilen Tamsayı tipteki DİZİ içindeki elemanların ORTALAMASINI geriye döndüren fonksiyon yazınız.

SORU: Kendisine gönderilen Tamsayı tipteki DİZİ içindeki elemanların ENKÜÇÜK değerini geriye döndüren fonksiyon yazınız.

SORU: Kendisine gönderilen Tamsayı tipteki DİZİ içindeki elemanların ENBÜYÜK geriyeye döndüren fonksiyon yazınız.

SORU: Bütün fonksiyonları Bir ANA MENÜ de birleştiren program

ANA MENÜ

1-Küçük Bul

2-Büyük Bul

3-Toplam Bul

4-Ortalama Bul

5-ÇIKIŞ

Ana programda yandaki gibi ekran çıktısı olacak.

Kullanıcı:

1 seçerse dizinin en küçük elemanını bulacak

2 seçerse dizinin en büyük elemanı bulacak

3- seçerse dizinin toplamını bulacak

4-seçerse ortalamasını

5 seçerse PROGRAM SONLANACAK

misal, 1 seçti dizinin en küçük elemanını buldu, tekrar aynı ekran karşısına çıkması lazım

5 seçene kadar program çalışmaya devam edecek

(KUCUKBUL, BUYUKBUL, TOPLAMBUL, ORTALAMABUL birer fonksiyon olarak oluşturulacak, geriye değer döndürecek.

SORU: TAMSAYI tipindeki Bir DİZİ içindeki ÇİFT / TEK sayıların adedini bulan fonksiyonu yazınız.

SORU ZOR: Kendisine gönderilen Tamsayı tipteki DİZİ içindeki TEK SAYILARI geriye döndüren fonksiyon

(NOT: Bu örnekte geriye birden fazla değer dönmesi gerekir. Bunun için dönüş değerinin DİZİ olması gerekir ancak Flowgorithm de bu mümkün değildir. Bir çözümü tabii ki vardır.)

Örnek:

Sayıları bir dizi içinde fonksiyona göndeririz.

Geriye dönüş olarak STRING tipte bir ifade seçeriz.

Sayıların karışmaması içinde araya VİRGÜL ekleriz.

Ana programda da VİRGÜL ifadesine göre sayıları ayıklayan SINIRLAYICI | DELİMİTTER olarak sayıları tekrar diziye aktarırız.

Örnek: Elimizde aşağıdaki gibi bir dizi olsun.

SAYILAR[5]	
0	11
1	20
2	37
3	50
4	67

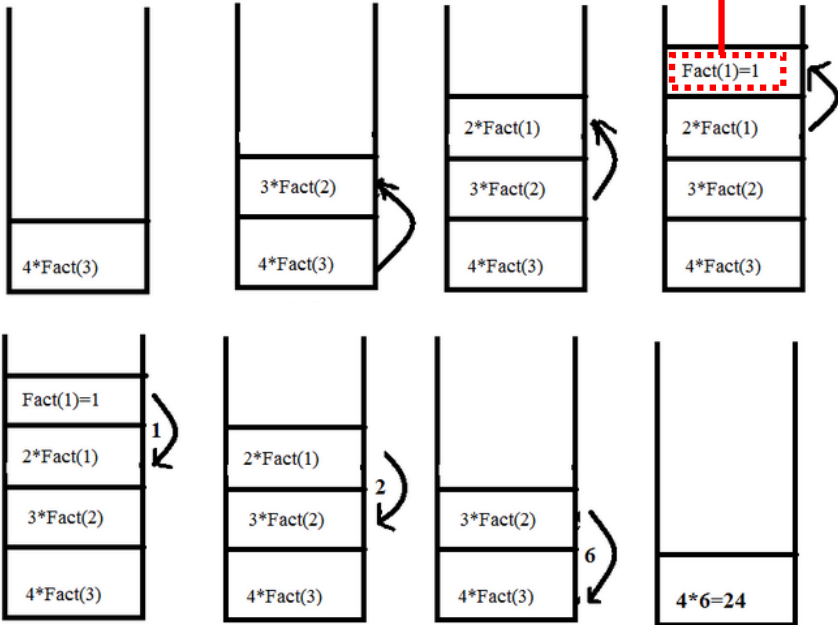
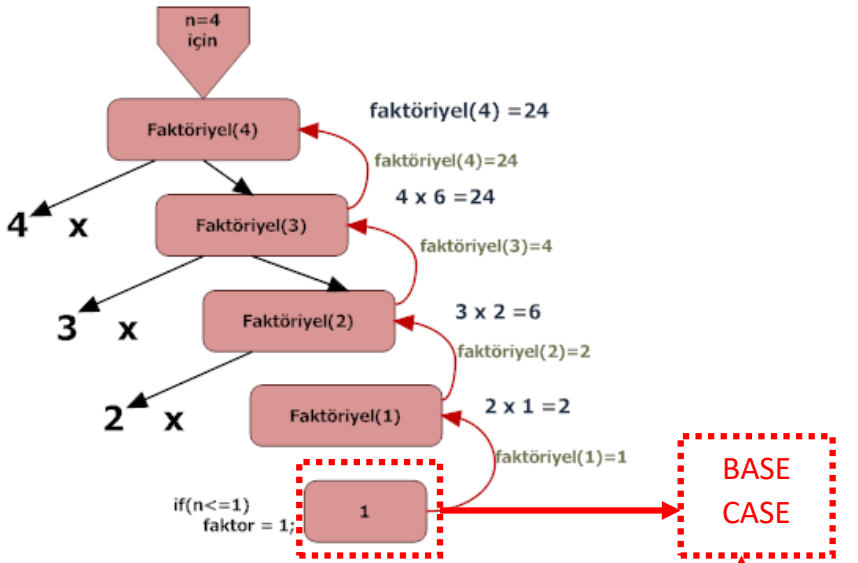
SONUC="11,37,67" olacak şekilde algoritma tasarlamak gerekir.

(ÖZYİNELEMELİ / RECURSİVE) FONKSİYON SORUSU

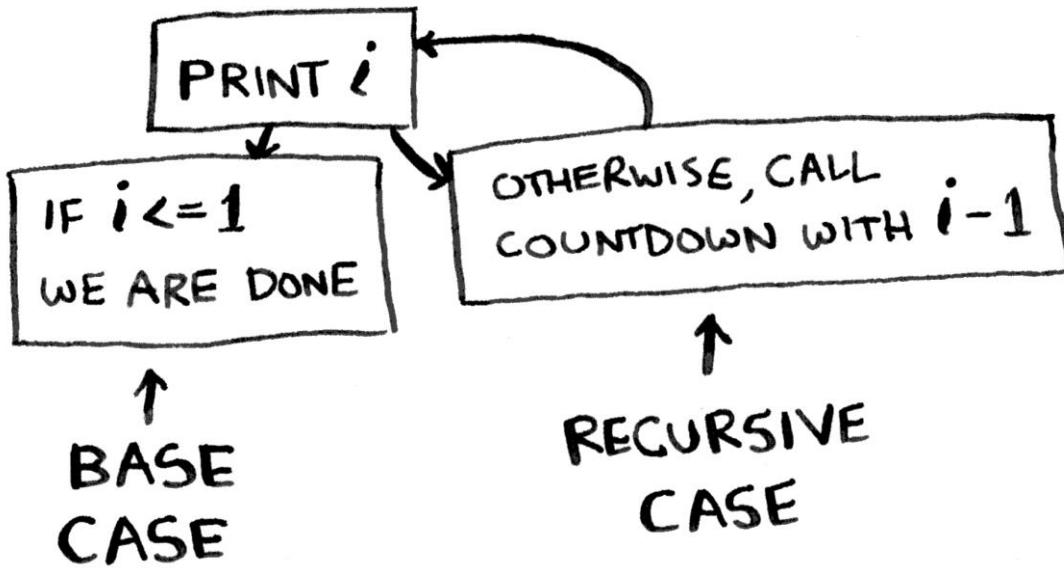
Bir fonksiyonun içinde yine kendisinden bir parça bulunuyorsa bu fonksiyonlara **ÖZYİNELEMELİ (RECURSİVE)** fonksiyon denir.

Açıklama [08]: RECURSİVE fonksiyonlar konusunu araştırınız.

Faktöriyel nedir? Araştırınız.



BASE-CASE: işlemin en son durması gereken adımı belirtir.
RECURSIVE-CASE: Tekrarlanması gereken durumlar.



SORU-BASİT: 1...n' e kadar olan pozitif sayıları toplayan recursive TOPLA fonksiyonu yazınız.

SORU-ORTA: Faktöriyel Hesaplama fonksiyonu (10.SINIF MAT)

FAKTÖRİYEL (AZCIK 10. Sınıf Matematik, fazla değil)

1'den başlayıp belirli bir sayma sayısına kadar olan sayıların çarpımına o sayının faktöriyeli denir. Ünlem işareti ile gösterilir.

$1! = 1$ dir. (merak eden araştırabilir.) (BASE CASE)

$$2! = 2 \times 1 = 2$$

$$3! = 3 \times 2 \times 1 = 6$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

$$n! = n \times (n-1)!$$

SORU-ZOR: 0 girilene kadar, girilen bir tamsayının 1'e kadar hep yarısını ekrana yazan program. (While olacak, recursive)
(TEK sayıların 2 ye bölümünde kalan kısmı ihmal edilecektir)
ÖRN: $17 / 2 = 8$ olacak şekilde

SORU ÖRNEKLERİ

1. Kelimeyi tersten yazma. (dünya → aynüd)
2. Girilen kelime, cümle, sayının PALİNDROM olup/olmadığını bulma ("kek" , 12321, "ey edip adanada pide ye")
3. Girilen kelimenin, aynı hecenin tekrarı ile oluşup oluşmadığını bulma (Kaykay, Mermer, Gaga ...)
4. Km/Mil, Kg/Gram, Döviz Çevirme işlemleri
5. abc 3 basamaklı bir sayı=> $abc = a^3 + b^3 + c^3$ olan sayıları bulan programı yazınız.
6. Kelimedeki Sesli/sessiz harf sayısını bulma.
7. Kelime karıştırma (rastgele yerlerini değiştirme) (bir sonraki soru için bunu fonksiyon olarak yazarsak, kullanışlı olur)
8. Kelimeyi anlamlı karıştırma (selam → sleam merhaba → mbhear)
9. Girilen bir sayının ASAL SAYI olup olmadığını bulma (asal sayı belirleme algoritması araştır)
10. Bir sayının asal çarpanlarını bulma

11. Ad Soyadı, düzgün eposta adresine dönüştürme
(Ömer ERMiŞ → omerermis@gmail.com)

12. 0-100 aralığında sayı tahmin oyunu

13. 2 kişilik, Girilen bir kelimenin son harfi ile biten yeni bir kelime türetme oyunu. Değilse uyaracak.
(kullanılan bir kelime tekrar kullanılamaz. Bunun için kurala uygun girilen kelimeleri, bir diziye ekleyip kontrol için orada saklanması lazım)
1->Elma 2->Ağaç 1->Çati 2->İrmak ...

14.

SORU-1: Kelimeyi Tersten Yazma

(Dizi kullanmadan ya da Dizi ile yapılabilir.) Bu işlemi bir fonksiyon olarak yazabiliriz, ileride kullanılması için)

Yapılacaklar:

1. Kullanıcıdan bir kelime iste
2. Uzunluğunu bul
3. Uzunluk kadar tekrar döngüsü kur.
 - a. Char() fonksiyonu kullan.
4. Çıktı üret

Bilgisayar → Uzunluk 10 karakter		
[0]	B	
[1]	i	
[2]	l	
[3]	g	
[4]	i	
[5]	s	
[6]	a	
[7]	y	
[8]	a	
[9]	r	

R

RA

RAY

RAYA

RAYAS

SORU-2: PALİNDROM kontrol algoritması.

(adana , kek, 12321 palindrom kelime/sayılarıdır)

Yapılacaklar:

1. Kelime iste
2. Kelimeyi tersten bir deęişkene ya da diziye aktar.
3. İlk girilen kelime ile yeni oluşturulan deęeri karşılaştır.
4. %100 olarak eşitlik varsa PALİNDROM dur, yoksa deęildir.

SORU-3: Girilen kelimenin aynı hecenin tekrarı olup olmadığını bulma.

(Mermer, kaykay ...)

(İPUCU: Kelime uzunluğu ????)

(Dikkat: BÜYÜK / küçük harf)

mermer	
0	M
1	e
2	r
3	m
4	e
5	r

→ 0 - 3
→ 1 - 4
→ 2 - 5

dede	
0	D
1	E
2	D
3	E

→ 0 - 2
→ 1 - 3