

## Programlamada En Çok Kullanılan İngilizce Kelimeler

Software = Yazılım	Front-end = Ön Yüz	Blog = Kişisel Sayfa
Byte = Bayt	Checkbox = İşaret Kutusu	Language = Dil
Default = Varsayılan	immutable = Sabit	insert = Ekleme
Data = Veri	Exit = Çıkış	Page = Sayfa
Data Type = Veri Türü	Option = Seçenek	Add = Ekleme
Random = Rastgele	Transition = Geçiş Efekt	False = Yanlış
Output = Çıktı	input = Veri Girişi	Print = Basmak
Loops = Döngüler	Pseudo Code = Kaba Kod	if = Eğer
Double = Çift	For = İçin	User = Kullanıcı
Algorithm = Algoritma	Browser = Tarayıcı	Cut = Kesmek
Continue = Devam Etmek	State = Durum	Font = Yazı Tipi
Class = Sınıf	Install = Kurmak	Cookie = Çerez
Password = Şifre	App = Uygulama	Save = Kaydetmek
Date = Tarih	integer = Tam Sayı	Shadow = Gölge
Setup = Kurmak	Network = Ağ	Variable = Değişken
Comment = Yorum	Array = Dizi	Break = Kırılma
Else = Değilse/ Yoksa	Start = Başlamak	Backup = Yedekleme
Button = Düğme	Delete = Silmek	Cancel = İptal Etmek

Database = Veritabanı	Folder = Klasör	Back = Geri
File = Dosya	Search = Aramak	Plug in = Eklenti
inheritance = Miras	About = Hakkında	Hardware = Donanım
Syntax = Söz Dizimi	True = Doğru	Settings = Ayarlar
Security = Güvenlik	Method = Metod	Tool = Araç
While = İken	Stop = Durmak	And = Ve Or = Veya

## Algoritma da Ve Programlama da Karşımıza Çıkabilecek Matematiksel İşlemler

% = Mod Alma (Bir sayının bölümünden kalanı bulmak için kullanılır. Bir sayının çift veya tek olduğunu bulmak için kullanabiliriz.)

### Karşılaştırma İfadeleri

> (Büyüktür) < (Küçüktür) >=(Büyük Eşittir) <=(Küçük Eşittir)  
= (Bir değer atamak için kullanılır ör: a = x + y -- Burada a nın değeri x+ y ye eşittir.)

Önermeler (Karşılaştırma yaparken kullanırız or(veya) and(ve) anlamına gelir. Ör: Karşılaştırmayı ve ile bağlarsak iki değeri de sağlaması gerekir veya dersek bir değeri karşılaması yeterlidir.)

## Yazılım Öğrenebileceğiniz Web Siteleri

W3schools (Birçok programlama dilinin konu anlatımları mevcuttur)  
Github(Örnek kodlar bulabileceğiniz bir yer)  
Stackoverflow(Hata ile karşılaşırsanız burada arama yapabilirsiniz)  
free-css(Ücretsiz birçok template ve layout bulabilirsiniz)  
Udemy(Ücretli-ücretsiz birçok alanda kurslar bulabilirsiniz)

[Youtube](#)(Birçok ücretsiz videolar bulabilirsiniz)  
[Codepen.io](#)(Burada Html-Css-Javascript ile ilgili örnekler bulabilirsiniz)  
[Draw.io](#)(Algoritma akış diyagramı çizeceğimiz web sitesi)  
[Canva](#)(Ücretsiz birçok tasarım aracı imkanı sunar. Ücretlisi de mevcut)  
[Color Hunt](#)(Ücretsiz renk seçimi yapabileceğiniz web sitesi)

## Yazılımda Bilmemiz Gereken Bazı Terimler

**Yazılım** = Elektronik aygıtların birbirleriyle haberleşmesi ve uyumlu bir şekilde haberleşmesi için gerekli makine kodlarıdır.(Tanım 1)

Yazılım elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir(Wikipedia tanımı)

**Donanım** = Bilgisayarları oluşturan fiziksel parçaların tümüne donanım denir. (ör: Anakart , Harddisk , Ram , Ekran Kartı)

**Kod** = Programların oluşması için gerekli olan verilerin tümüne kod denir.

**Veritabanı** = Birbirleri ile bağlı bilgilerin depolandığı yerlere veritabanı denir.

**Front-end** = Bir web sitesine girdiğimizde karşımıza çıkan her şeydir .  
Ör: button , menüler

**Back-end** = Bir web sitesinin görünmeyen kısmıdır. Yani arka tarafta çalışılan kısımdır. Sunucu tarafında çalışan kısım da denilebilir.

Bir örnekle açıklamak gerekirse . Bir giriş ekranında görünen her şey butonlar inputlar ön yüz giriş yaptığımız da çalışan kısım ise arka yüz olarak tanımlayabiliriz.

## Çalışma Ortamının Kurulması

Algoritma akış diyagramı çizerken birçok farklı uygulama web siteleri kullanabilirsiniz. Aynı şekilde metinsel olarak ifade ederken Word gibi metin editörlerini kullanabilirsiniz. Ben akış diyagramlarını çizerken Draw.io'yu kullanacağım. Kısaca kullanımından bahsedelim.

- İlk öncelikle web siteye giriş yapın.
- İlk karşınıza çıkan ekranda diyagramları nerede kaydetmek isteyeceğinizi soruyor . (Tercih size kalmış ben bilgisayara kaydediyorum)
- Seçtikten sonra yeni bir diyagram oluşturmak istediğiniz kısmı seçin. Diğer kısım ise daha önce yapmış olduğunuz bir akış diyagramını açmak için kullanılır.
- Boş bir diyagram seçip isim vererek akış diyagramını oluşturmuş oluruz.
- Sol taraftaki şekiller ile akış diyagramını oluşturabiliriz.
- Sağ tarafta ise renk ve yazı tipi ile ilgili değişiklikler yapabiliriz.

## Algoritma Oluşturmanın Püf Noktası

Algoritmayı metinsel olarak ifade ederken ilk öncelikle soruyu bölümlere ayırmamız gerekir. Ve bu şekilde ilerlersek soruyu daha kolay bir şekilde çözebilir. Bunu bir örnekle hemen açıklayalım.

Ör: Girilen iki sayının/ toplamını veren/ ve bunu ekrana yazdıran /programın aşamalarını yazalım.

Cevap: Algoritmayı başlatırız .İlk öncelikle girilen iki sayı dediğimiz an durup hemen iki sayıyı tanımlarız . Ardından toplamını veren diyor orada duralım . Şimdi iki sayının toplamını nasıl buluruz onu yazalım( $\text{toplam} = x+y$ ) ve bir değişkene( geçici değere) atayalım burada değişken toplam olur. Ardından bunu ekrana yazarız. Ardından algoritmayı sonlandırırız. Bu şekilde sorulara yaklaşırsanız daha basit bir şekilde çözebilirsiniz.

**\*\* Bu bölümde algoritmaya giriş yapmadan önce öğrenmemiz gereken bazı şeylerden bahsettim.**

## --Algoritma--

**Algoritma** = Bir amaca ulaşmak için izlenmesi gereken yola denir.(1)  
Bir sorunu veya problemi çözmek için izlenen sıralı adımlara denir(2)

Ör: 2 tane araba şirketi düşünelim. Bunlar A ve B firması olsun.

A => İşlerini tasarlayıp çizerek mühendislerden ve tecrübeli kişilerden fikir alarak ilerleyen bir firma olsun

B=> Sıralı adımlar izlemeyen yetkili kişilerden fikir almayan kendi kafasına göre hareket eden bir firma olsun.

A=> Tasarım ve fikir alışverişi yaptığı için yaklaşık işe başlaması 2 hafta sürdü.

B=> Tasarım ve fikir alışverişi yapmadığı için 2 haftadır işin içinde ancak çizim tasarım gibi izleyeceği yolu oluşturmadığı için hatalar ve yanlışlar yaptı. Ne kadar erken başlasada sıralı adımlar izlemediği için işin süresi ve maliyeti uzar.

Sonuç=>B firmasının yaptığı araç güvenlik ve tasarımsal açıdan sağlıklı bir araç olmaz. İşte algoritma tam burada devreye girer .

Problemin Çözümü : Araba Üretmek

Problem : Araba Üretimi

Araba Yapımı İçin Algoritmalar : Projeler , Çizimler , Planlar

**\*\* Programlamada da aynı şekilde en önemli kısım işe başlamadan önce amaca ulaşmak için algoritma hazırlamaktır.**

Ör: Yukarıdaki örnekten devam edelim. A firması işi 2 yılda teslim eder. B firması ise işi 4 yılda teslim eder. Çünkü amacı için herhangi bir ön hazırlık yapmadığı için hatalar alır ve iş süresi uzar buda maliyete yansır.

**\*\*** Programlama da algoritmayı hazırladıktan sonra bunu programlama diline dökeriz. Burada programlama dilinin önemi yoktur. Önemsiz derken bunu hemen bir örnekle açıklayalım.

Ör: Bir otel için bir mobil uygulama işi geldiğini düşünelim. Siz bu uygulamayı yaptıktan sonra karşınızdaki kişi bunun hangi programlama dili ile yapıldığını sorgulamaz. Sadece programın nasıl işlediğine yani performansına bakar.

### --Algoritma Adımları Nasıl Olmalıdır--

Algoritma hakkında biraz kafamızda bir şeyler oluştu ancak algoritma adımlarını nasıl yazacağımız hakkında pek bir fikrimiz yok . Bu bölümde algoritma adımlarını yazarken nasıl bir yol izlememiz gerekiyor onlardan bahsedelim.

**Performans** = Ürünün performansı olarak tanımlayabiliriz. Yani ürünün çalışırkenki hızı , kapladığı alan gibi performans kriterlerine uyum sağlamalıdır.

**Sonlu Olmalı** = Algoritmanın bir başlangıç noktası ve bir bitiş noktası olmalıdır. Eğer bunlar olmazsa sonsuz döngü dediğimiz kavram ortaya çıkar.

Ör: A firması sipariş alarak üretim yapmaya devam eder. Böylece bir başlangıç ve bitiş noktası olur.

B firması ise sipariş almadan bir sayı olmadan üretim yapar. Bir bitiş noktası olmaz . Ve böyle devam ederse B firması iflas etmeye doğru ilerler. Yani maddi bakımdan zarar eder.

**Anlaşılır Olmalı** = Yazdığımız algoritma anlaşılır olmalıdır. Bir örnekle bunu açıklayalım.

Ör: Oyun yapmak için bir takımımız olduğunu düşünelim . Kod kısmındaki arkadaşımız maaşını alıp gruptan ayrılmıştır. Diğer kod yazan arkadaşlar o koda baktığında hepsi aynı şeyi anlıyor ise o kod anlaşılır olmuştur. Eğer farklı farklı cevaplar gelirse o kod anlaşılır değildir.

**Kesin Olmalı** = Yaptığımız algoritma kesin olmalıdır . Yani her cevap geldiğinde cevap hep aynı olmalıdır.

Ör: Haftanın günlerini gösteren bir uygulama olduğunu düşünelim . 1 girince pazartesi 2 girince salı bu şekilde devam eden bir program olsun. Test aşamasında her seferinde 1 yazıldığı zaman pazartesi cevabı alınırsa kesin olmuş demektir. Ama 1 yazınca salı değerini alıyorsa kesin olmamış demektir.

## Günlük Hayattan Algoritma Örnekleri

### Market Alışverişi Algoritması

- 1- Evdeki eksikleri kontrol et
- 2- Eksikleri not al
- 3- Markete git
- 4- Alışveriş için sepet al
- 5- Eksikleri al ve sepete koy
- 6- Kasaya git
- 7- Ödeme yap
- 8- Alışverişi bitir ve eve dön

### Araba Sürme Algoritması

- 1- Arabanın lastiklerini kontrol et

- 2- Arabanın kaputunu aç
- 3- Yağ seviyesini kontrol et
- 4- Yağ yeterli ise kaputu kapat
- 5- Arabanın kapısını aç
- 6- Koltuğa otur
- 7- Kapıyı kapat
- 8- Koltuğu kendine göre ayarla
- 9- Aynaları kontrol et
- 10- Emniyet kemerini tak
- 11- Vitesin boşta olduğunu kontrol et
- 12- Kontağı çevir
- 13- Debriyaja bas ve vitesi 1e al
- 14- Bir ayağın frende iken el frenini indir
- 15- Hafifçe frenden ayağını kaldır ve hafifçe gaz ver
- 16- Arabayı sür

### --Algoritma İfade Şekilleri--

Algoritma iki şekilde ifade edilir. (Metinsel olarak // Pseudo Code(Kaba Kod -- Sözde Kod))

#### Metinsel Olarak İfade Edilmesi

<>Her satıra numara verilir. (1- / 1 / Adım 1: / 10- gibi)

<>Adım adım yazılır.

<>Başla ile başlar son ile biter. (Eş anlamlı ifadeler de kullanılabilir. Başlamak için start bitirmek için stop dur bitir gibi ifadeler de kullanılabilir)

**Ör:** Klavyeden girilen sayının iki katını alıp ekrana yazdıran programın metinsel olarak algoritmasını yazınız.

Adım 1: Başla

Adım 2: Bir sayı gir(sayı)



Adım 3: Sayıyı 2 ile çarp(cevap = sayi \* 2)

Adım 4: Yaz cevap (cevap)

Adım 5: Stop

/\* Açıklama:

Her zaman algoritmanın bir başlangıcı olmalıdır. (Adım 1:)

Bir sayı gireriz . Parantez içindeki sayi kod kısmında değişkenlere denk geliyor. Burada ise geçici isim olarak düşünebiliriz.(Adım 2:)

Sayıyı sorudaki denildiği gibi 2 ile çarpıyoruz ve buda cevap geçici ismine eşitlenir. Yani cevap dediğimiz şey sayının 2 ile çarpımı demektir.

(Adım 3:)

4. adımda ise cevap değişkenini ekrana yazdırmış oluruz.(Adım 4:)

Stop diyerek algoritmayı sonlandırırız.(Adım 5:)

\*/

### Sözde Kod (Pseudo Code)

<> Algoritmayı günlük konuşma dili ile ve programlama dili ile yazılmasına denir.

**Ör:** Kısa ve uzun kenarının değerini alarak bir dikdörtgenin alanını sözde kod şeklinde yazınız.

Adım 1: Dikdörtgenin kısa ve uzun kenarı değerini al

Adım 2: Alan = kısıkenar \* uzunkenar

Adım 3: Alan değerini ekrana yazdır

/\* Açıklama:

Kısa ve uzun kenarın değeri alınır. Bu işlemi 2 adımda da yapabiliydik . Yani ilk önce kısa sonra uzun kenarın değerini alabiliydik. Ancak minimum ve yalın bir şekilde yapmamız gerektiği için bu şekilde yapıldı.(Adım 1:)

Alan deęiřkenine (geici isme) kısıkenar \* uzunkenar deęerini vermiř olduk.(Adım 2:)

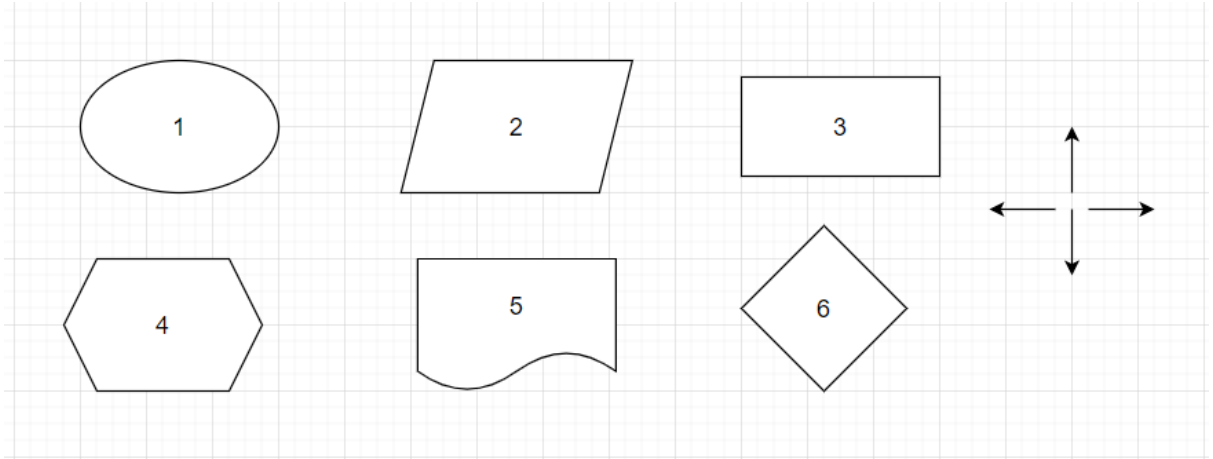
Son olarak ekrana Alan deęerini yazdırır.(Adım 3:)

\*/

**Not:** Bu soruyu arkadaşlarınız ile birlikte yapmaya alışın oęunuzun metinsel olarak ifade ederek farklı ifade edebilir . Ancak mantık her zaman aynı olacaktır .

### --Algoritma Akıř řeması--

Algoritmanın řekiller ile ifade edilmesine akıř řeması(diyagramı) denir.



1) Bařla ve bitir komutu iin kullanılır. (Elips)

- 2) Veri girişı için kullanılır . Parantez içinde yazdığımız sayı buna bir örnektir. (Paralelkenar)
- 3) Aritmetik işlemler için kullanılır. İki sayının toplamı buna bir örnektir.(Dikdörtgen)
- 4) Döngüler için kullanılır.(Altıgen)
- 5) Ekranı çıktı vermek için kullanılır.(Belge / Document)
- 6) Karşılaştırma için kullanılır . (Baklava dilimi / elmas / diamond )
- 7) Oklar ise akış yönünü belirler.

**Ör:** Klavyeden girilen 2 sayının toplamını veren programın metinsel olarak ifadesini ve akış diyagramını oluşturunuz.

Adım 1: Start

Adım 2: İki sayı gir(x,y)

Adım 3: x ve y sayılarını toplam ve toplama eşitle(toplam = x + y)

Adım 4: Yazdır toplam

Adım 5: Stop

/\* Açıklama: Metinsel olarak ifade

Start ile algoritma başlat(Adım 1:)

İki sayı tanımlarız . Bunlar x ve y dir. (Adım 2:)

İki sayıyı toplayıp toplam değerine atarız. Yani toplamın değeri x ve y nin toplamına eşittir.(Adım 3:)

Ekranı toplam ı yazdırır. Yani x+ y değerini yazdırır. (Adım 4:)

Stop ile algoritma bitirilir.

\*/

/\*Açıklama: Akış Diyagramı

Start diyerek algoritma başlatılır(Elips) (Algoritma başlangıcı)

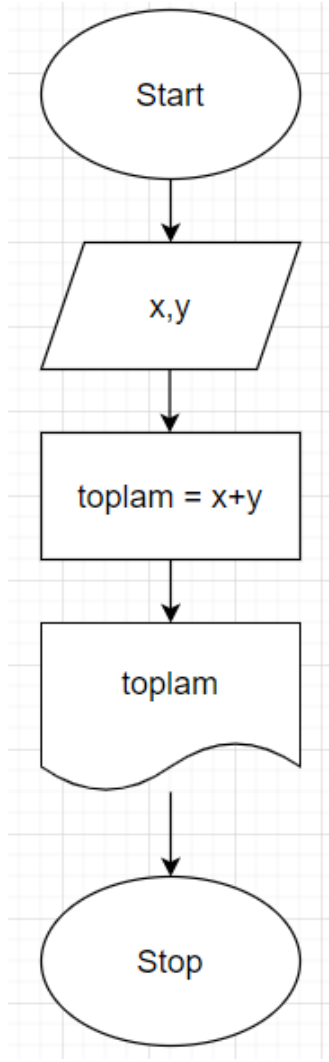
x ve y tanımlanır. (Paralelkenar)(Veri girişi)

$x+y$  değeri toplama eşitlenir . (Dikdörtgen) (Aritmetik işlem)

toplam değeri ekrana bastırılır. (Document / belge) (ekrana çıktı verme)

Stop diyerek algoritma sonlanır.(Elips)

\*/



Ör: Klavyeden girilen bir sayının pozitifmi negatifmi olduğunu bulan programın algoritmasını ve akış diyagramını oluşturunuz.

Adım 1: Başla

Adım 2: Bir sayı gir(sayı)

Adım 3: Eğer sayı < 0 ise Yaz “sayı negatif”

Adım 4: Değilse Yaz “sayı pozitif”

Adım 5: Dur

/\* Açıklama:

Başla ile algoritma başlatılır.(Adım 1:)

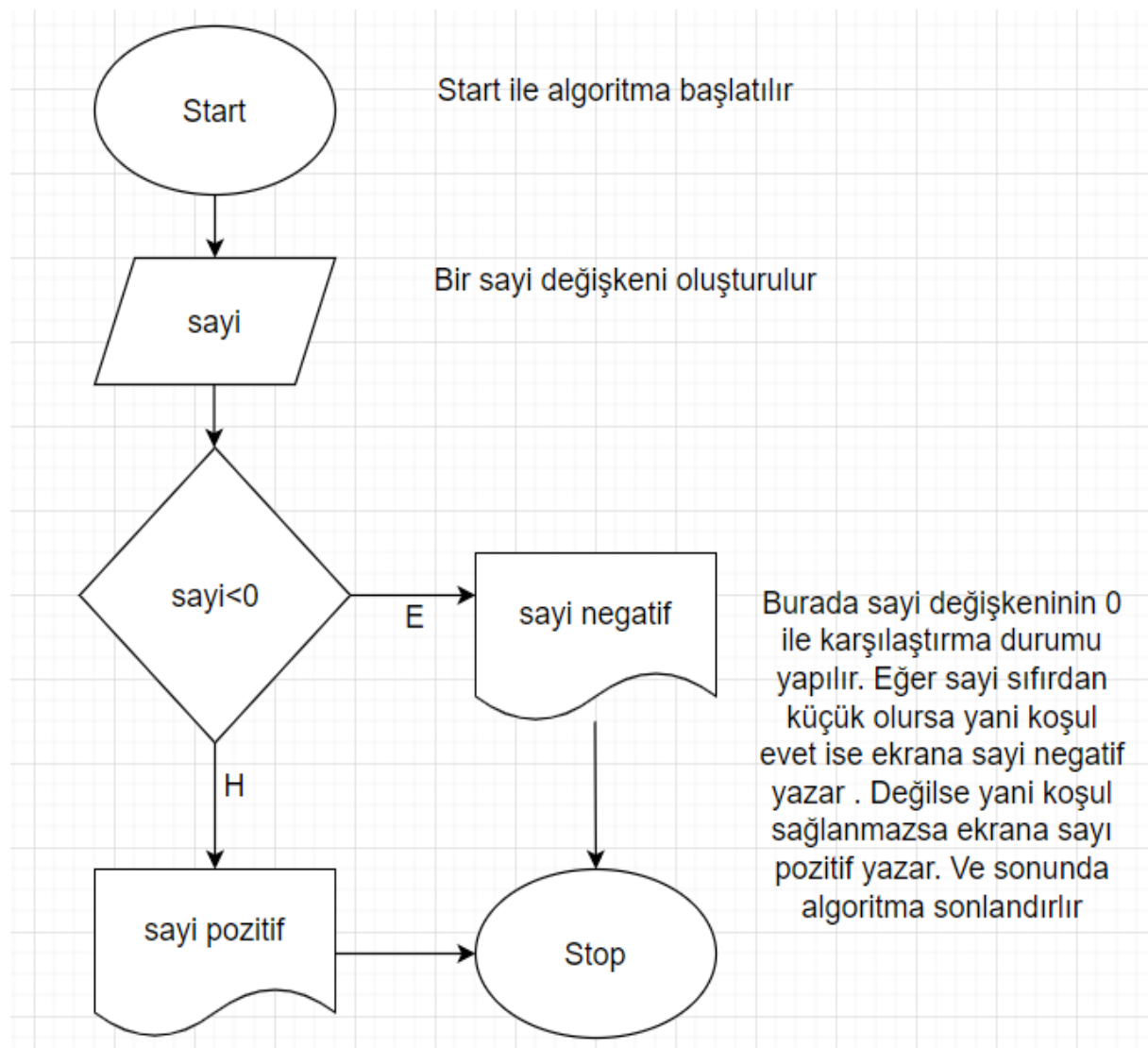
Bir sayı girişi olur yani tanımlanır (Adım 2:)

Bir karşılaştırma yapılır sayı sıfırdan küçükse ekrana sayı negatif yazar(Adım 3:)

Değilse yani sayı sıfırdan küçük koşulu sağlamaz ise ekrana sayı pozitif yazar(Adım 4:)

Dur ile algoritma sonlandırılır.(Adım 5:)

\*/



Ör: Klavyeden girilen 2 sayıdan büyük olanını ekrana yazdıran programın metinsel olarak ifadesini ve akış diyagramını oluşturunuz.

Adım 1: Başla

Adım 2: İki sayı gir (sayi1 , sayi2)

Adım 3: Eğer sayi1 > sayi2 Yaz sayi1

Adım 4: Değilse Yaz sayi2

Adım 5: Bitir

/\* Açıklama

Başla ile algoritma başlatılır(Adım 1:)

İki sayı girilir ve bunların adı sayi1 ve sayi2 dir.(Adım 2:)

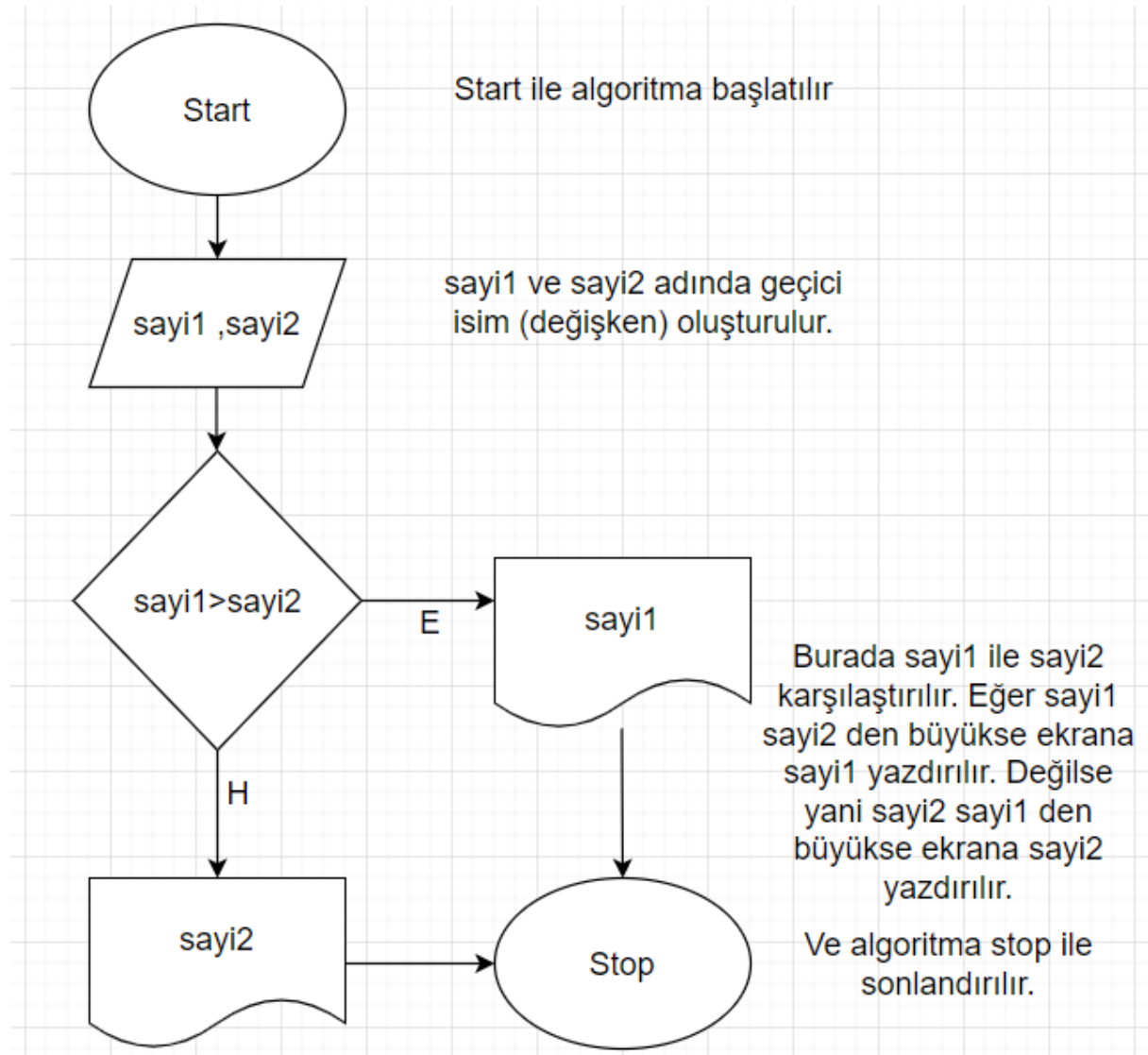
sayi1 sayi2 den büyükse ekrana sayi1 yazdırılır(Adım 3:)

Değilse yani sayi1 sayi2 den küçükse ekrana sayi2 yazdırılır.(Adım 4:)

Bitir ile algoritma sonlandırılır(Adım 5:)

\*/

Not : Daha farklı durumlarda eklenebilir. Sayının sıfıra eşit olması gibi. Şuanlık kafanızın karışmaması için oralara giriş yapmadım.



\*\* Bir sonraki örnekte döngüler konusuna giriş yapacağım. Döngüler sürekli tekrar gerektiren olaylar gerektiği zaman kullanılır. Alt alta sayıların yazdırılması bu duruma bir örnek olur. Birazdan yapacağım 2 örnekte ilki döngü kullanmadan ikincisi ise döngü kullanarak bir örnek yapacağız . Ve böylece döngülerin kullanım alanını daha iyi anlayacağız.



Ör: Ekran 4 kez “Merhaba” yazdıran programın metinsel olarak algoritmasını ve akış diyagramını oluşturunuz.

Adım 1: Başla

Adım 2: Yaz “Merhaba”

Adım 3: Yaz “Merhaba”

Adım 4: Yaz “Merhaba”

Adım 5: Yaz “Merhaba”

Adım 6: Son

/\* Açıklama

Başla ile algoritma başlatılır(Adım 1:)

Ekran 4 Merhaba yazılır 4 defa(Adım 2: Adım 3: Adım 4: Adım 5:)

Son ile algoritma sonlandırılır.

Döngü kullanmadan yaptığımız bu soru kullanımı yanlış değil ama eğer 1000 defa deseydi tek tek yapmak çok zor olurdu o yüzden bir sonraki soruda daha doğru olan kullanımını göreceğiz.

\*/

Ör: Ekran 4 kez “Merhaba” yazdıran programın metinsel olarak algoritmasını ve akış diyagramını oluşturunuz.

Adım 1: Başla

Adım 2: Sayac = 0

Adım 3: Yazdır “Merhaba”

Adım 4: Sayac = Sayac + 1

Adım 5: Eğer Sayac < 4 ise Adım 3 e git

Adım 6: Değilse Adım 7 ye git

Adım 7: Stop

/\*Açıklama:

Başla ile algoritma başlatılır(Adım 1:)

Sayaç bizim burada sayıları takip etmemizi sağlar. Başlangıç değeri sıfırdır. (Adım 2:)

Ekrana “Merhaba” yazdırır.(Adım 3:)

Ve sayacımıza bir eklenir (Adım 4:)

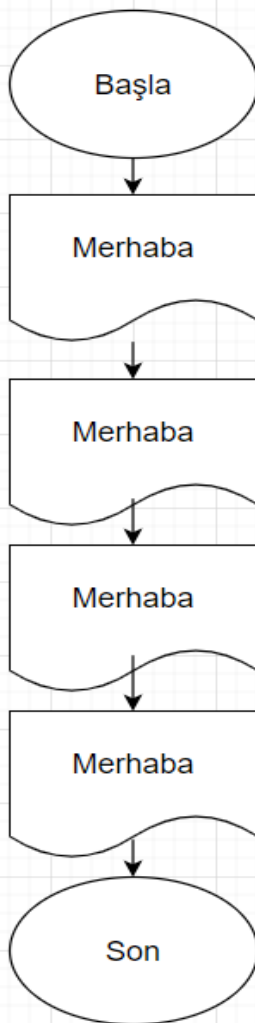
Eğer sayac değeri 4 den küçük ise adım 3 e döner ve tekrar ekrana “Merhaba yazar” .(Adım 5:)

Eğer sayac 4 den küçük değilse adım 7 ye gider.(Adım 6:)

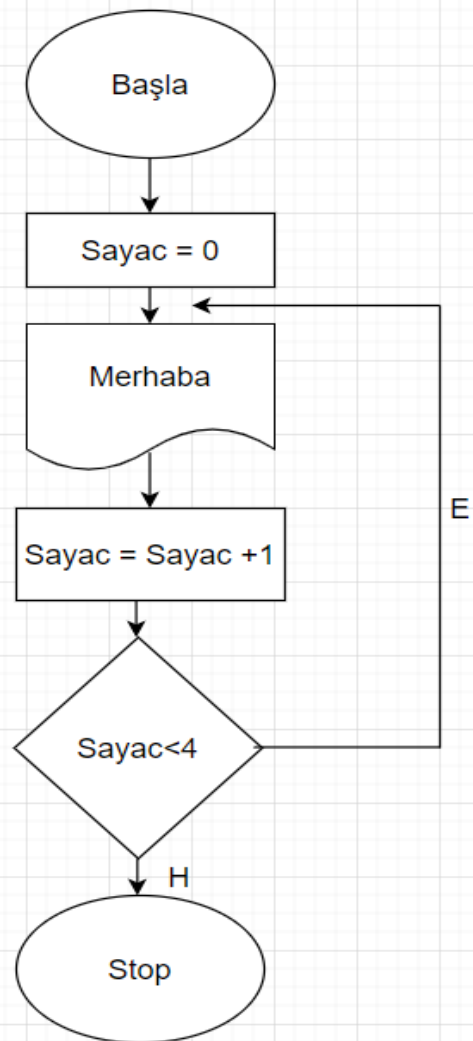
Algoritma Stop ile sonlanır.(Adım 7:)

\*/

Döngü Kullanılmadan Yapılan Örnek



Döngü Kullanılarak Yapılan Örnek



Ör: 1 ile 5 arasındaki sayıları toplayan ve ekrana yazdıran programın metinsel ifadesini ve akış diyagramını oluşturun.

Adım 1: Başla

Adım 2: Sayac = 1 , Toplam = 0

Adım 3: Toplam = Toplam +sayac

Adım 4: Sayac = Sayac + 1

Adım 5: Eğer sayac < = 5 ise Git adım 3

Adım 6: Değilse Yaz Toplam

Adım 7: Bitir

/\* Açıklama:

Başla ile algoritma başlatılır(Adım 1:)

Sayac adında değişken oluşturulur ve değeri 1 dir. Toplam adında bir geçici veri (değişken) oluşturulur ve değeri 0 dır.(Adım 2:)

Toplam ın değeri toplam + sayac olur. Yani Sayac her arttığında toplama eklenir. (Adım 3:)

Sayac değeri bir arttırılır . Toplam değerine eklendikten sonra yeni değer için Sayac bir arttırılır.(Adım 4:)

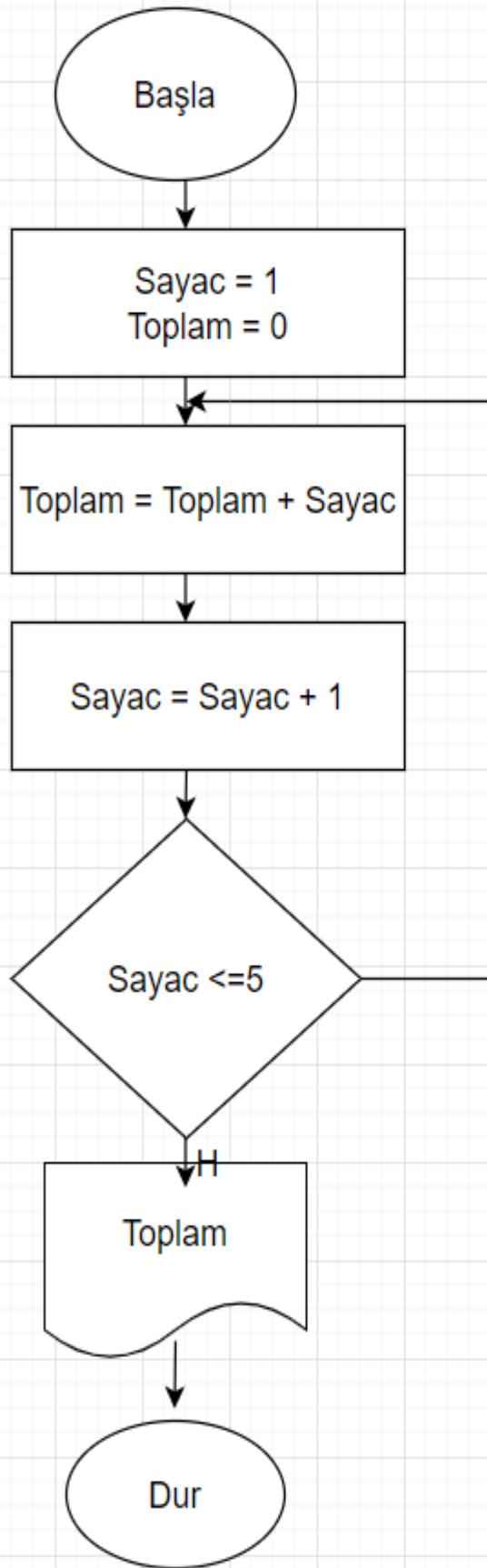
Eğer sayac değeri 5'ten küçük eşitse adım 3 e dönülür.(Adım: 5)

Değilse yani sayac 5 den büyükse ekrana Toplam değerini yazdırır.(Adım 6:)

Bitir ile algoritma sonlandırılır.(Adım 7:)

Buradaki Toplam dediğimiz şey aslında Sayac ile birer birer arttırdığımız sayıları topladığımızdaki verilerin tutulduğu yerdir.

\*/

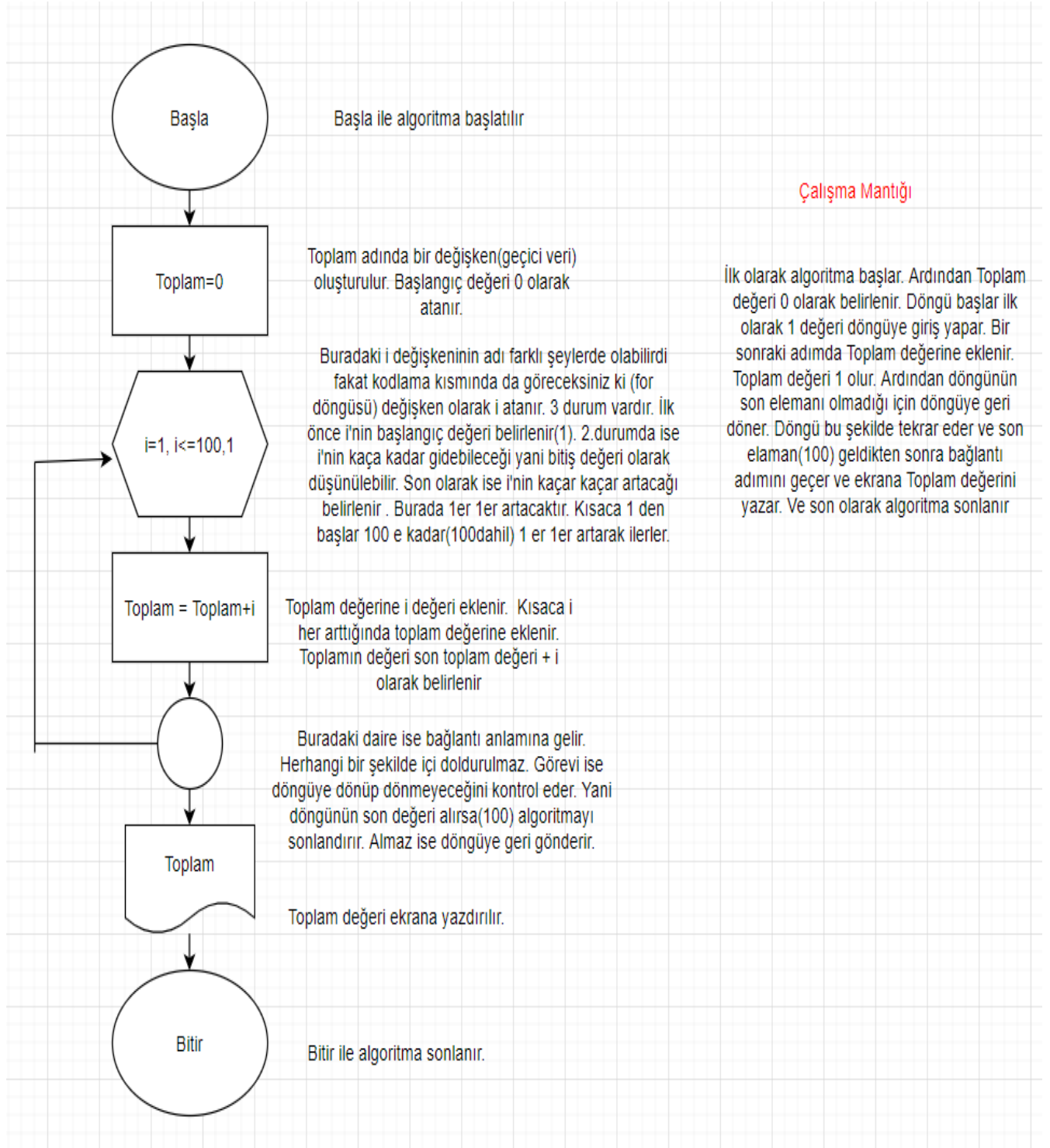


Burada Toplam dediğimiz  
Sayaçtaki sayıların bir bir  
arttırılarak toplanıp sonucun  
yazıldığı kısımdır

E Sayac ise sayıların bir  
bir arttırılıp yazıldığı  
kısmıdır.

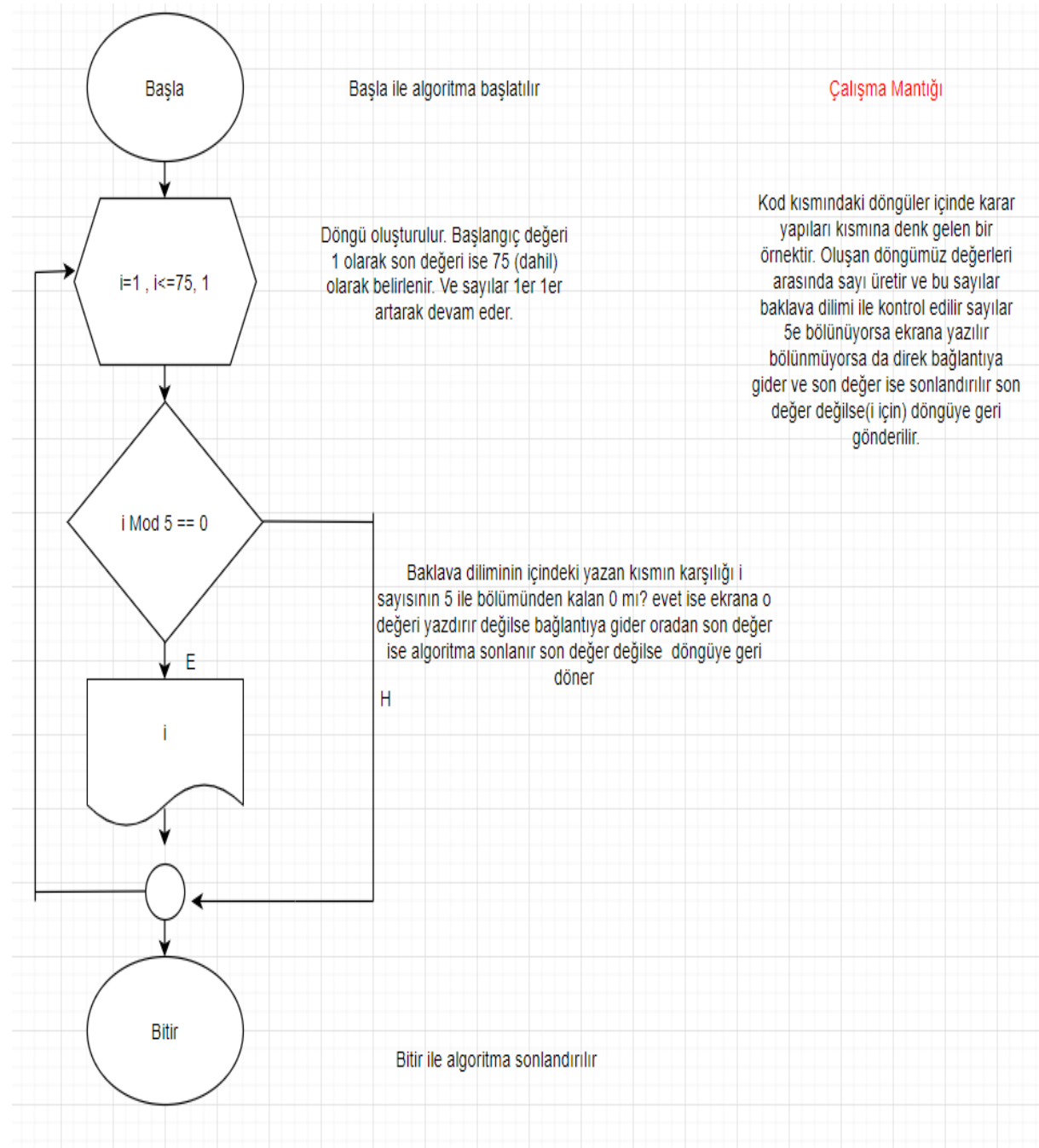
Burada sayının 5 den  
küçük eşit olduğunu  
kontrol ediyor. Evet ise  
Adım 3 gidiliyor. Değilse  
ise Ekrana Toplam  
yazdırılıyor. Ve algoritma  
sonlandırılıyor

Ör: 1 - 100 arasındaki sayıların toplamını bulup ekrana yazdıran programın akış diyagramını çizin.



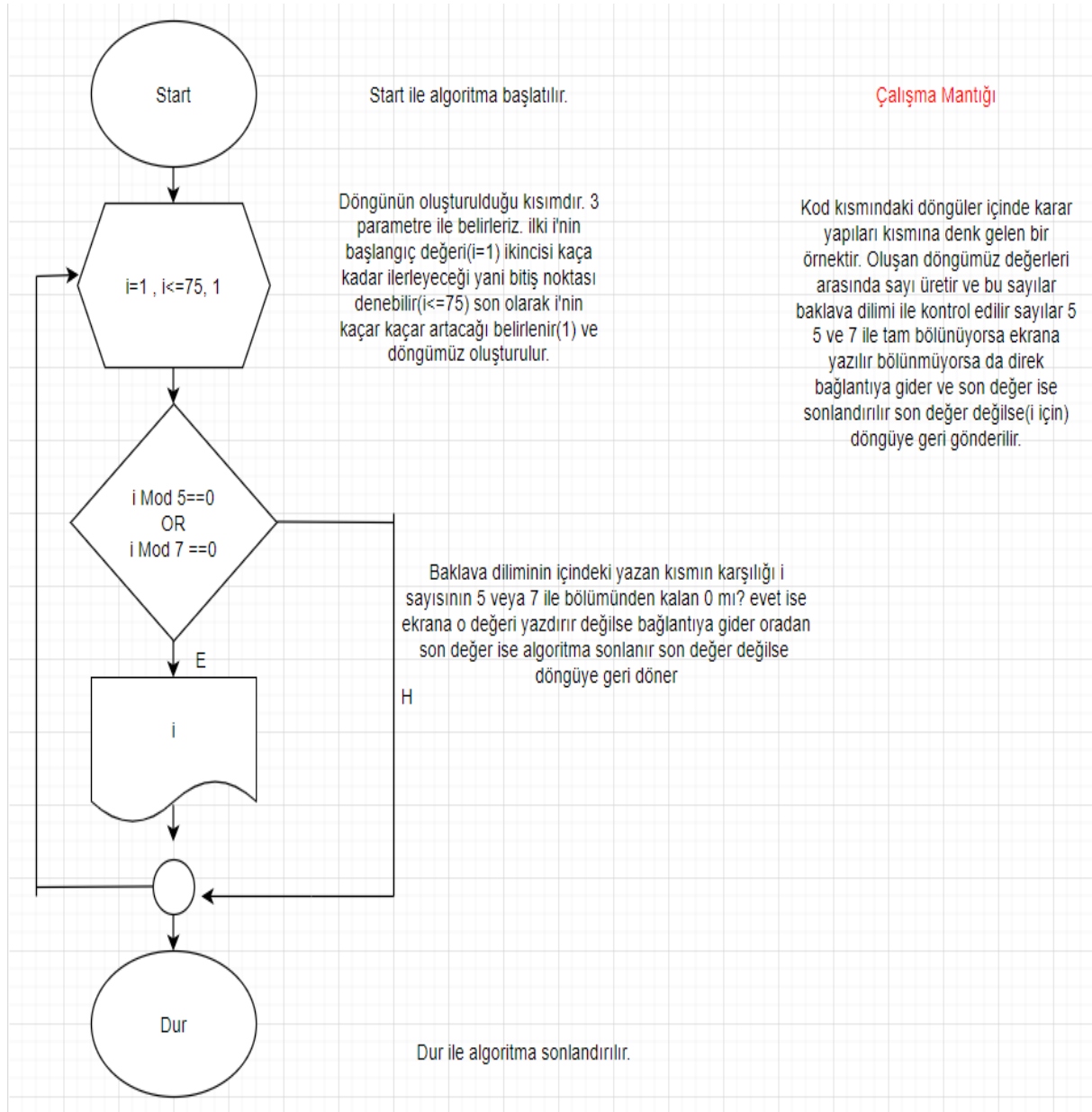
\*\*\*\*\*Yandaki açıklamalar algoritmaya dahil değildir. Ben onları not olarak yazdım siz çizerken yanlarına bu şekilde not yazmayınız.

Ör: 1 - 75 arasında 5e tam bölünen sayıların akış diyagramını çiziniz.



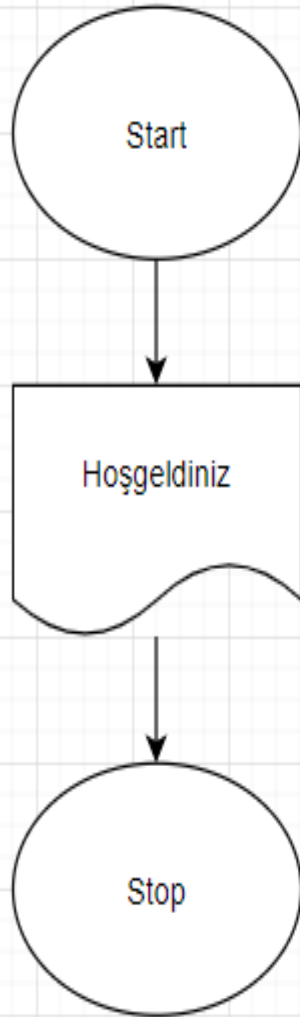
\*\*\* Bu örneğimizde tek bir sayıyı kontrol ettik şimdi sıradaki örneğimizde iki sayı istenirse nasıl yapacağımızı göreceğiz . (Veya / ve ile bağlayarak sorunu çözebiliriz. Ve dersek her iki koşulu sağlaması gerekir yani 2 ve 3 bölünen sayılar denirse her ikisinde bölünmesi gerekir veya denirse bir tanesine bile bölünse koşul sağlanır kısaca farkı bu şekilde ifade edebiliriz.)

Ör: 1 - 75 arasında 5e veya 7e tam bölünen sayıların akış diyagramını çiziniz.



\*\* Burada ve ile de bağlayabilir dik. Eğer öyle bir soru gelseydi yani 2 sayıya da bölünmesi istenseydi AND kullanılırdı. OR yerine AND yazıp bağlayabilir dik.

Ör: Ekran “Hoşgeldiniz” yazan programın metinsel olarak adımlarını ve akış şemasını oluşturunuz.(Basit bir soru olsada soru çeşitlerini görmeniz adına bu soruyu koydum)



Adım 1: Start

Adım2: Yaz "Hoşgeldiniz"

Adım 3: Stop



Ör: 90 dan 0a (dahil değil) geriye doğru sayan ve sayıları ekrana bastıran program'ın metinsel olarak algoritma adımlarını ve akış diyagramını çiziniz.

Adım 1: Start

Adım 2: Sayi = 90

Adım 3: Eğer sayi < 0 Git adım 6

Adım 4: Değilse Sayi dan bir çıkar ve sonucu Sayi değişkenine ata.  
(Sayi = Sayi-1)

Adım 5: Yaz Sayi Git Adım 3

Adım 6: Stop

/\* Açıklama

Start ile algoritma başlatılır(Adım 1:)

Sayi değişkeni oluşturulur ve değeri 90 olur.(Adım 2:)

Eğer sayi 0 dan küçük ise adım 6 ya gidilir yani algoritma sonlanır(Adım 3:)

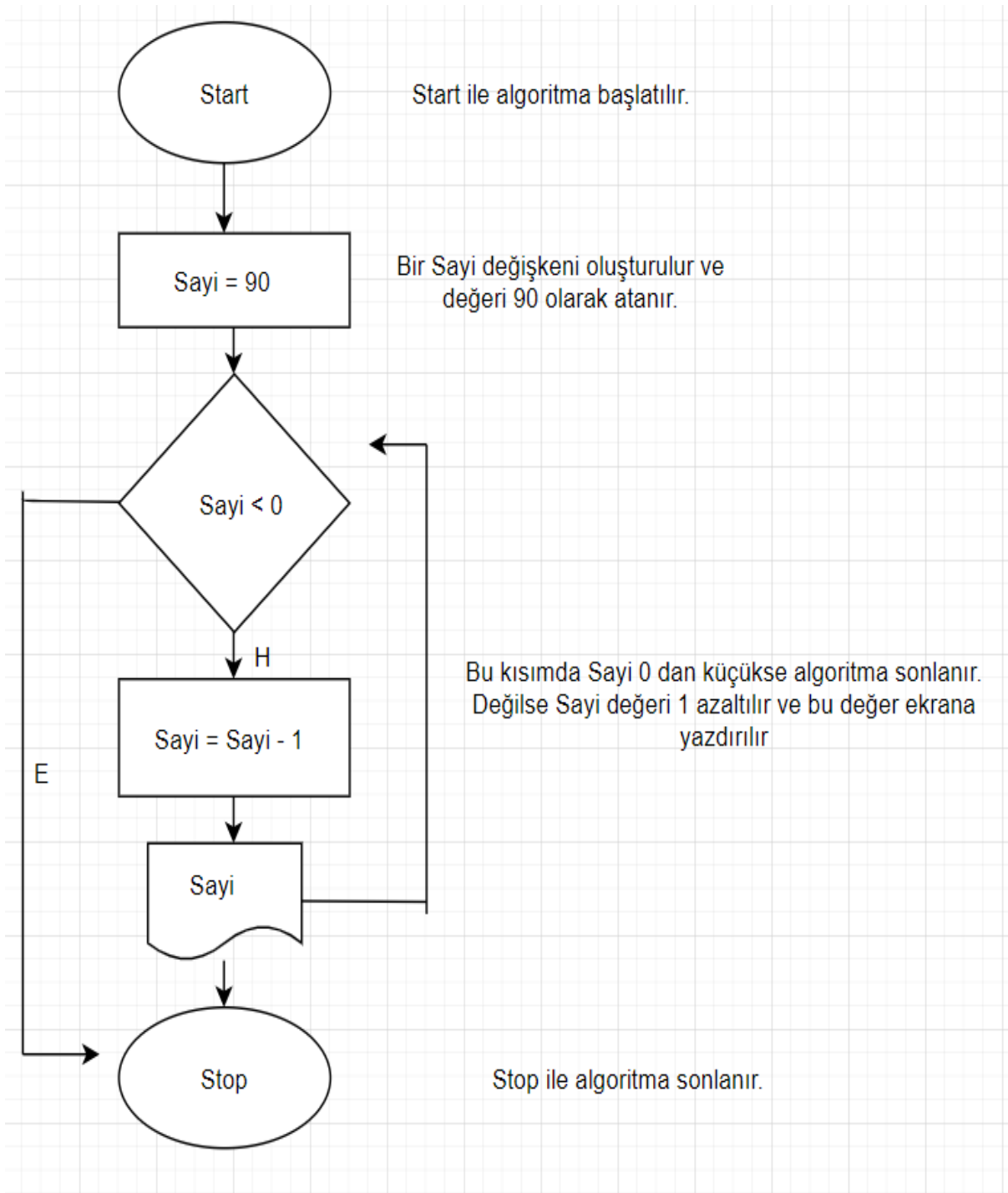
Eğer sayi 0 dan küçük değil ise Sayi dan bir çıkarılır ve bu değer Sayi değişkenine atanır.(Adım 4:)

Ekrana Sayi değerini yazar. Ve adım 3 e gider yani döngüye geri döner.(Adım 5:)

Stop ile algoritma sonlandırılır(Adım 6:)

\*/

Akış diyagramı da aşağıdaki gibidir.



\*\* Eğer sıfırdan 90'a doğru ekrana yazdıracak olsaydık başlangıç değeri 1 olurdu ve baklava dilimin içi( Sayi > 90) şeklinde olurdu. Ve Sayi değerine bir eklenirdi(diğer örnekte çıkardık). Bu şekilde bu tarz bir soruyuda çözebilirsiniz.

Ör: Klavyeden yaşı girilen bir kişinin ehliyet alıp alamayacağını hesaplayan programın algoritma adımlarını ve akış diyagramını oluşturunuz.

Adım 1: Start

Adım 2: Yaşı gir (Yas)

Adım 3: Eğer yas  $\geq$  18 Yaz "Ehliyet Alabilir"

Adım 4: Değilse Yaz "Ehliyet Alamaz"

Adım 5: Stop

/\* Açıklama:

Start ile algoritma başlatılır(Adım 1:)

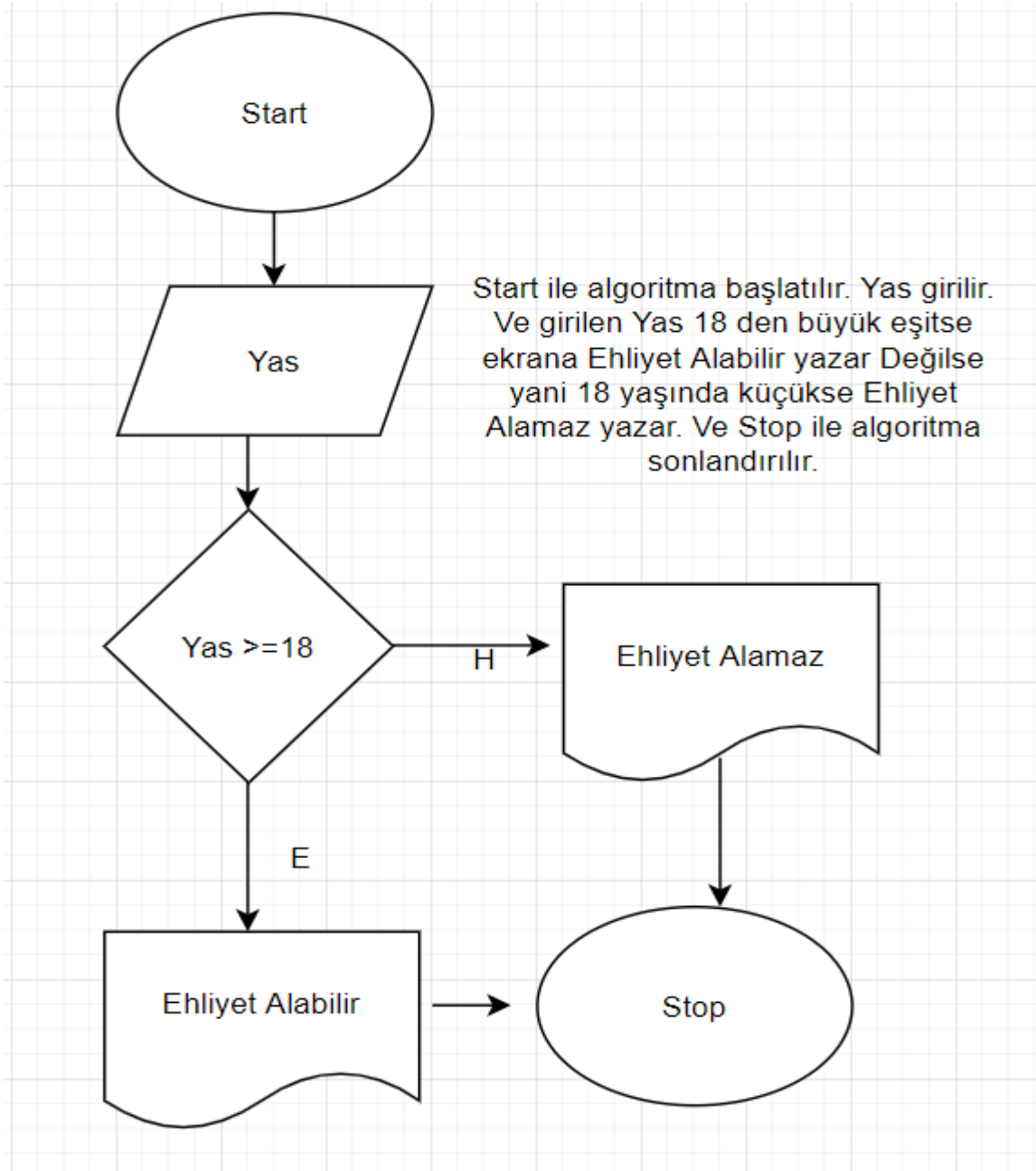
Yaş girilir geçici isim olara Yas dedik.(Adım 2:)

Eğer yas 18 den büyük eşit ise ekrana Ehliyet Alabilir yazar.(Adım 3:)

Değilse yani Yas 18 den küçükse ekrana Ehliyet Alamaz yazar(Adım 4:)

Stop ile algoritma sonlandırılır(Adım 5:)

\*/



\*\* Değişkenlere isim verirken TÜRKÇE karakter kullanmamaya dikkat edelim. Çünkü kod kısmı için gerekli olacak. Elimden geldiğince Algoritmayı anlatmaya çalıştım. Fikir ve düşünceleriniz benim için çok değerli yorum kısmında belirtirseniz bir yanlışım hatam varsa bende görmüş olurum. Sınavınızda başarılar dilerim :).