

BMG-Final

## Soru 1:

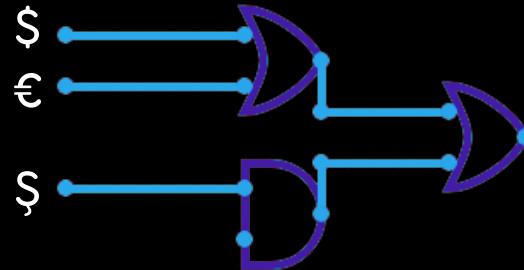
A) Aşağıdaki karakterlerin decimal (onluk) değerleri verilmiştir. Bu değerleri kullanarak hexadecimal (16'lık) binary (2'lik) ve octal (8'lik) karşılıklarını bulun.

	Karakter	Decimal
1	Euro (€)	8364
2	Dolar (\$)	36
3	₺	350

1. Euro (€)
2. Dolar (\$)
3. "₺" harfi

B) üç karakterin (€, \$, ₺) kaç bit ve kaç byte yer kapladığını hesaplayınız.

C) Her adımda mantık kapısının çıktısını bulun.



adg

## Soru 1.A cevap:

1. Öncelikle verilen karakterlerin decimal (onluk) değerlerini belirleyelim. Nasıl yapacağınızı sitede gösterdim.

Euro (€) → 8364

Dolar (\$) → 36

Ş harfi → 350

2. Decimal'den Hexadecimal (16'lık) Sisteme Çevirme: (Yapılış şekli sitede mevcut)

- Euro (€) → 0x20AC
- Dolar (\$) → 0x24
- Ş harfi → 0x15E

3. Decimal'den Binary (2'lik) Sisteme Çevirme: (Yapılış şekli sitede mevcut)

- Euro (€) → 10000010101100<sub>2</sub>
- Dolar (\$) → 100100<sub>2</sub>
- Ş harfi → 10101110<sub>2</sub>



#### 4. Decimal'den Octal (8'lik) Sisteme Çevirme(Yapılış şekli sitede mevcut)

Euro (€) → 20254<sub>8</sub>

Dolar (\$) → 44<sub>8</sub>

Ş harfi → 536<sub>8</sub>

##### 1.A) Sonuç :

Karakter	Decimal (10'luk)	Hexadecimal (16'lık)	Binary (2'lik)	Octal (8'lik)
Euro (€)	8364	0x20AC	10000010101100 <sub>2</sub>	20254 <sub>8</sub>
Dolar (\$)	36	0x24	100100 <sub>2</sub>	44 <sub>8</sub>
Ş harfi	350	0x15E	101011110 <sub>2</sub>	536 <sub>8</sub>



## Soru 1.B cevap:

Bilgisayarda karakterler Unicode (UTF-8 veya UTF-16 gibi) sistemlerle saklanır. Her karakterin bellekte kapladığı alan değişebilir.

Karakter	Decimal (Onluk)	UTF-8 Bellekte Kapladığı Alan
Euro (€)	8364	3 byte (24 bit)
Dolar (\$)	36	1 byte (8 bit)
Ş harfi	350	2 byte (16 bit)

Bellekte saklanma şekline göre:

1 byte = 8 bit

Karakter	Kaç Byte?	Kaç Bit?
Euro (€)	3 byte	24 bit
Dolar (\$)	1 byte	8 bit
Ş harfi	2 byte	16 bit



## Soru 1.B cevap:

### Toplam Byte ve Bit Hesaplama

Toplam Byte:

$3+1+2=6$  byte

Toplam Bit:

$3*8 + 2*8 + 1*8 = 48$  bit

A handwritten signature in white ink, consisting of stylized cursive letters, likely 'adg'.

## Soru 1.C cevap:

Mantık kapıları binary (0 ve 1) değerlerle çalıştığı için önce karakterlerin binary değerlerini kullanıyoruz:

Karakter	Decimal	Binary
Euro (€)	8364	10000010101100 <sub>2</sub>
Dolar (\$)	36	100100 <sub>2</sub>
Ş harfi	350	101011110 <sub>2</sub>

**Not:** Mantık işlemlerinde genellikle en az 8 bit kullanılır. Dolar (\$) sadece 6 bit olduğu için başına 0 ekleyerek 8 bit yapıyoruz:

Dolar (\$) → 00100100<sub>2</sub>



## Soru 1.C cevap:

OR kapısı en az bir giriş 1 ise çıkış da 1 olur.

İşlemi bit bit yapalım:

Euro (€) → 10000010101100<sub>2</sub>

Dolar (\$) → 00000000100100<sub>2</sub>

-----  
Sonuç → 10000010101100<sub>2</sub>

Binary Sonuç: 10000010101100<sub>2</sub> (İsterse soruda bu rakamı sonra Hexadecimal ve Octal'a çeviririz.)

Hexadecimal: 2OAC<sub>16</sub>

Octal: 20254<sub>8</sub>





## Soru 1.C cevap:

**Note:** AND kapısı yalnızca girişteki tüm değerler 1 olduğunda 1 sonucunu üretir. Eğer en az bir giriş 0 ise çıkış da 0 olur.  
AND kapısı tek giriş için kullanıldığında çıkış aynısı olur (çünkü kendiyi AND işlemine girer):

Ş Harfi  $\rightarrow 101011110_2$

-----

Sonuç  $\rightarrow 101011110_2$

Binary Sonuç:  $101011110_2$

Hexadecimal:  $15E_{16}$

Octal:  $536_8$



## Soru 1.C cevap:

Son olarak, ilk OR sonucuyla AND sonucunu OR (VEYA) kapısına sokuyoruz:

OR Çıkışı (Euro \$)  $\rightarrow 10000010101100_2$

AND Çıkışı (\$)  $\rightarrow 00000101011110_2$

-----

Sonuç  $\rightarrow 10000111111110_2$

Binary Sonuç:  $10000111111110_2$

Hexadecimal:  $21FE_{16}$

Octal:  $20776_8$



## Soru 1.C cevap:

Sonuç :

İşlem	Binary	Hexadecimal	Octal
OR (VEYA) – Euro (€) ve Dolar (\$)	10000010101100 <sub>2</sub>	0x20AC	20254 <sub>8</sub>
AND (VE) – Ş Harfi	101011110 <sub>2</sub>	0x15E	536 <sub>8</sub>
Final OR (VEYA) – İlk iki sonucu birleştirme	10000111111110 <sub>2</sub>	0x21FE	20776 <sub>8</sub>

Final Cevap :

Euro (€) OR Dolar (\$) → 10000010101100<sub>2</sub> (0x20AC, 20254<sub>8</sub>)

Ş Harfi AND → 101011110<sub>2</sub> (0x15E, 536<sub>8</sub>)

Final OR (VEYA) → 10000111111110<sub>2</sub> (0x21FE, 20776<sub>8</sub>)



Mantıksal ve matematiksel konularla ilgili notları siteye geçici olarak paylaştım. PDF'in bu bölümünden sonra sözel konulara yönelik sorular yer alacaktır. Hocanın paylaştığı PDF'leri de çalışabilirsiniz.

\_\_\_\_\_X\_\_\_\_\_ Web uygulamalarında sunucu tarafında çalışan, veri işleme, veritabanı yönetimi ve iş mantığını yöneten yazılım geliştirme alanıdır.

cevap: Backend

---

\_\_\_\_\_X\_\_\_\_\_ özellikle dinamik web siteleri ve web uygulamaları geliştirmek için kullanılan sunucu taraflı (server-side) bir programlama dilidir. Web sitelerinin arka planında çalışarak verileri işler, veritabanlarıyla iletişim kurar ve kullanıcıya uygun içerik sunar.

cevap: PHP      açılımı : Hypertext Preprocessor

PHP'nin Kullanım Alanları

- ✓ Dinamik Web Sayfaları Geliştirme
- ✓ Kullanıcı Girişi ve Yetkilendirme (Login/Signup)
- ✓ Form İşleme ve Veri Doğrulama
- ✓ Veritabanı Yönetimi (CRUD İşlemleri)
- ✓ E-Ticaret Siteleri ve Ödeme Sistemleri
- ✓ API Geliştirme (RESTful, SOAP)
- ✓ Otomasyon ve Arka Plan İşlemleri



\_\_\_\_\_X\_\_\_\_\_ özellikle dinamik web siteleri ve web uygulamaları geliştirmek için kullanılan sunucu taraflı (server-side) bir programlama dilidir. Web sitelerinin arka planında çalışarak verileri işler, veritabanlarıyla iletişim kurar ve kullanıcıya uygun içerik sunar.

cevap: HTML açılımı: (Hyper Text Markup Language)

---

### CSS açılımı: (Cascading Style Sheets)

Amaç: HTML ile oluşturulan web sayfalarının tasarımını ve görünümünü şekillendirmek

### SQL açılımı: (Structured Query Language)

Amaç: veritabanlarını yönetmek ve sorgulamak

---

Backend ile ilgili diller nelerdir?

PHP, Python, Java, Node.js, Ruby

Frontend ile ilgili diller nelerdir?

HTML, CSS, JavaScript, TypeScript, React



## Programlama dillerinin nesilleri

1. Nesil: Makine Dili
  2. Nesil: Assembly
  3. Nesil: C, Java, Python, Pascal, Ruby
  4. Nesil: SQL, MATLAB, R, Prolog, Perl
  5. Nesil: Lisp, Prolog, Mercury, OPS5, Wolfram
- 

## MAC

Media Access Control (Medya Erişim Kontrolü), bir ağ arayüzüne ait fiziksel adresleme sistemidir. ve benzersiz bir kimlik numarasıdır.

A stylized, handwritten signature in white ink, located in the bottom right corner of the slide. The signature appears to be a cursive representation of a name, possibly 'Adil'.

\_\_\_\_\_X\_\_\_\_\_ İlk bilgisayar programcısı olarak kabul edilen ve Charles Babbage'ın Analitik Motoru için algoritmalar geliştiren matematikçidir.

Cevap: Ada Lovelace (1815-1852)

---

İlk Bilgisayar: ENIAC (1945) (Tam adı: Electronic Numerical Integrator and Computer (ENIAC))

Dünyanın ilk genel amaçlı elektronik bilgisayarı olarak kabul edilir. Amerikan Ordusu için balistik hesaplamalar yapmak amacıyla geliştirildi. 18.000 vakum tüpü kullanıyordu ve devasa büyüklükteydi. Programlanması zordu; kabloların fiziksel olarak değiştirilmesi gerekiyordu.

---

İlk Programlama Dili: Assembly & Fortran

---

RAM, CPU ve RAM türleri:

**RAM (Random Access Memory):** Geçici bellek türü, işlemlerin hızlı yürütülmesini sağlar.

**CPU (Central Processing Unit):** Merkezi işlem birimi, bilgisayarın temel işlemlerini yapar.

**RAM Türleri:** DDR, DDR2, DDR3, DDR4, DDR5





\_\_\_\_\_X\_\_\_\_\_, cihazların internet ve yerel ağlarda birbirleriyle iletişim kurmasını sağlayan benzersiz sayısal adresleme sistemidir.

📌 IP Adresi:

Her cihaza atanmış bir kimlik numarasıdır.

Veri paketlerinin hedef cihaza ulaşmasını sağlar.

İki tür IP adresi vardır: IPv4 ve IPv6.

cevap ve açılımı: IP (Internet Protocol)

IP Adresinin Formatı ve Türleri:

◆ IPv4 (Internet Protocol Version 4)

32-bit uzunluğundadır (4 bloktan oluşur).

0-255 arasında 4 sayı içerir.

Örnek: 192.168.1.1

◆ IPv6 (Internet Protocol Version 6)

128-bit uzunluğundadır (8 bloktan oluşur).

Hexadecimal (16'lık) sayı sistemi kullanır.

Örnek: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Tür	Açıklama
Public (Genel) IP	İnternet üzerinde herkese açık bir adres.
Private (Özel) IP	Yerel ağda kullanılan, dışarıdan erişilemeyen IP.
Static (Statik) IP	Değişmeyen sabit IP adresi.
Dynamic (Dinamik) IP	Modem her bağlandığında değişen IP adresi.



X

\_\_\_\_\_ internet üzerindeki verilerin iletimini sağlayan protokoldür.

HTTP, web tarayıcınız ile web sunucuları arasındaki iletişimi sağlar.

İstemci (client) ve sunucu (server) arasında veri transferini yönetir.

**Cevap :** HTTP (Hyper Text Transfer Protocol – Hiper Metin Transfer Protokolü)

1. Kullanıcı tarayıcısına bir URL yazar (örn: <http://example.com>).
2. Tarayıcı, sunucuya bir istek (request) gönderir.
3. Sunucu, isteğe uygun sayfayı tarayıcıya yollar.
4. Tarayıcı, gelen veriyi işler ve kullanıcıya gösterir.

Örnek HTTP URL'si:

<http://www.google.com>

HTTP (HyperText Transfer Protocol – Hiper Metin Transfer Protokolü)

### HTTP'nin Temel Özellikleri

- ✓ İçeriği Web Sayfalarında Gösterir: HTML, CSS, JavaScript dosyalarını iletir.
- ✓ Bağlantısızdır: Her istekte yeni bir bağlantı kurulur.
- ✓ Metin, resim, video ve veri aktarımını sağlar.
- ✓ HTTP, şifrelenmemiştir, bu yüzden güvenli değildir!



\_\_\_\_\_X\_\_\_\_\_ HTTP'nin daha güvenli bir versiyonudur ve SSL/TLS şifreleme protokolü ile çalışır.

**Cevap :** HTTPS (Hyper Text Transfer Protocol Secure)

HTTPS veri transferi sırasında bilgileri şifreleyerek üçüncü tarafların bu verilere ulaşmasını engeller.

#### HTTPS'in Avantajları

- ✓ Şifrelenmiş İletişim: Veriler SSL/TLS ile şifrelenir, böylece hacker'lar tarafından okunamaz.
- ✓ Güvenli Kimlik Doğrulama: Kullanıcılar gerçek bir web sitesine bağlandığından emin olabilir.
- ✓ SEO Avantajı: Google ve diğer arama motorları HTTPS kullanan siteleri üst sıralara çıkarır.
- ✓ Veri Bütünlüğü: Veri iletimi sırasında verilerin değiştirilmesini önler.

Örnek HTTPS URL'si:

`https://www.google.com`



## HTTP ve HTTPS Arasındaki Farklar:

Özellik	HTTP	HTTPS
Güvenlik	Güvenli değil	SSL/TLS şifreleme kullanır
Bağlantı	Düz metin olarak iletilir	Şifrelenmiş veri iletimi sağlar
Hız	Daha hızlı	Şifreleme nedeniyle biraz daha yavaş
SEO Etkisi	Google tarafından önerilmez	SEO açısından daha avantajlı
URL Başlangıcı	http://	https://



BMG-Final Son Kare

## 1. Bilgisayar Ağları ve Protokoller

Protokol: Cihazların birlikte çalışabilmesi için belirlenen kurallar bütünüdür.

TCP/IP: En yaygın kullanılan internet protokolü. TCP (Transmission Control Protocol) ve IP (Internet Protocol)'den oluşur.

MAC Adresi: Bir cihazın ağ içindeki fiziksel adresidir (48 bit, değiştirilemez).  
(media access control address)

IP Adresi: İnternette cihazların tanımlanmasını sağlar (IPv4: 32 bit, IPv6: 128 bit).  
(Internet Protocol Address)

## 2. OSI Modeli (Open System Interconnection)

7 katmandan oluşur ve her biri farklı görevleri yerine getirir:

1. Fiziksel Katman: Veri iletiminin kablolar ve sinyaller aracılığıyla yapılmasını sağlar.
2. Veri Bağlantı Katmanı: MAC adresleriyle cihazlar arası iletişimi düzenler.
3. Ağ Katmanı: IP adresleriyle paket yönlendirme yapar.
4. Ulaşım (Transport) Katmanı: TCP ve UDP protokolleri burada çalışır.
5. Oturum Katmanı: İki cihaz arasındaki bağlantıyı yönetir.
6. Sunum Katmanı: Verinin şifrelenmesi ve sıkıştırılması burada yapılır.
7. Uygulama Katmanı: Kullanıcıya en yakın katmandır (HTTP, FTP gibi protokoller burada çalışır).

## 3. TCP/IP Modeli

OSI modeline göre daha pratik bir modeldir ve 4 katmandan oluşur:

Uygulama Katmanı: Kullanıcı ile doğrudan iletişim kurar (HTTP, FTP).

Taşıma Katmanı: TCP ve UDP kullanarak veri iletimini yönetir.

İnternet Katmanı: IP paketlerini yönlendirir.

Ağ Erişim Katmanı: Fiziksel veri iletimini sağlar.

## 4. IP ve Ağ İletişimi

IP Paketleri: Başlık bilgisi ve veri kısmından oluşur.

ICMP Protokolü: Ağ hatalarını bildirmek için kullanılır (Ping, Traceroute gibi komutlar burada çalışır).

DDoS Saldırıları: Ağa aşırı istek göndererek sistemleri çökertmek için yapılan saldırılardır.

## 5. İnternet Temelleri

HTTP & HTTPS: Web sayfalarının aktarım protokolleri.

DNS (Domain Name System): Alan adlarını IP adreslerine çevirir.

FTP (File Transfer Protocol): Dosya aktarımı için kullanılır.

## 6. Veritabanı ve Büyük Veri

Veritabanı: Düzenli veriler topluluğudur.

SQL (Structured Query Language): Veritabanlarını yönetmek için kullanılır.

Büyük Veri (Big Data): Büyük boyutlu veri kümelerini işleyip analiz etmeyi sağlar.

## 7. NoSQL Veritabanları

Belge Tabanlı Veritabanı (Document-Based): JSON veya XML formatında veri saklar (MongoDB).

Anahtar-Değer (Key-Value) Veritabanı: Veri anahtar-değer çifti olarak saklanır (Redis).

Grafik Tabanlı Veritabanı: Düğüm ve kenarlar ile ilişkileri yönetir (Neo4j).



## 8. Bilgisayar Ağlarında Cihazlar ve Yapılar

- Switch (Anahtar): Veri paketlerini MAC adresine göre yönlendirir. Hedefe doğrudan gönderir.
- Hub: Gelen veriyi tüm bağlı cihazlara yollar (verimsizdir).
- Router (Yönlendirici): IP adresine göre ağlar arası veri yönlendirmesi yapar.
- Gateway (Geçit): Farklı ağ protokolleri arasında dönüşüm yapar.

## 9. Ağ Güvenliği ve Saldırı Türleri

- DDoS (Distributed Denial of Service): Birden fazla cihazdan yapılan yoğun trafik saldırısıdır.
- MITM (Man in the Middle Attack): İki taraf arasındaki iletişimi gizlice dinleyerek verileri ele geçirir.
- Phishing (Kimlik Avı): Sahte e-postalar ve web siteleri kullanarak kullanıcı bilgilerini çalmaya yönelik bir saldırdır.
- Brute Force Attack: Şifreleri tahmin etmek için sürekli denemeler yapan saldırıdır.
- Firewall (Güvenlik Duvarı): Zararlı trafiği engellemek için kullanılır.

## 10. Web Teknolojileri

- Web Tarayıcıları: Google Chrome, Mozilla Firefox, Safari gibi internet sitelerini açan programlardır.
- Web Sunucuları: Web sayfalarını barındıran sistemlerdir (Apache, Nginx, IIS).
- HTML (Hypertext Markup Language): Web sayfalarını oluşturmada kullanılan dil.
- CSS (Cascading Style Sheets): Web sayfalarının tasarımını yapar.
- JavaScript: Web sayfalarına etkileşim kazandırır.

### 1. Veri İletimi ve Şifreleme Yöntemleri

- Asimetrik Şifreleme (Public Key Encryption): Açık ve özel anahtar kullanılarak güvenli iletişim sağlar (Örn: RSA).
- Simetrik Şifreleme: Aynı anahtar kullanılarak veri şifrelenir ve çözülür (Örn: AES, DES).
- Hashing (Özetleme): Veriyi tek yönlü şifreleyerek güvenli hale getirir (Örn: SHA, MD5).

## 12. İşletim Sistemleri ve Bellek Yönetimi

İşletim Sistemi (OS - Operating System): Bilgisayar donanımı ve yazılımını yönetir (Windows, Linux, macOS).

Bellek Yönetimi:

RAM (Rastgele Erişimli Bellek): Geçici veri depolama alanıdır, hızlıdır.

ROM (Salt Okunur Bellek): Kalıcıdır, sistemin temel fonksiyonlarını içerir.

Cache (Önbellek): Hızlı erişim için kullanılan küçük bir bellek türüdür.

Virtual Memory (Sanal Bellek): RAM yetersiz kaldığında disk alanı kullanılarak oluşturulur.

## 13. Algoritmalar ve Programlama Temelleri

Algoritma: Problemleri çözmek için adım adım verilen talimatlar dizisidir.

Programlama Dilleri:

Düşük Seviye: Makine dili, Assembly.

Yüksek Seviye: Python, Java, C++, JavaScript.

Veri Yapıları:

Dizi (Array): Aynı türden elemanları içeren veri yapısıdır.

Bağlı Liste (Linked List): Dinamik hafıza kullanımına sahip veri yapısıdır.

Yığın (Stack): LIFO (Last In, First Out) prensibiyle çalışır.

Kuyruk (Queue): FIFO (First In, First Out) prensibiyle çalışır.

Ağaç (Tree): Hiyerarşik veri yapısıdır.

## 14. Yazılım Geliştirme Süreçleri

SDLC (Software Development Life Cycle): Yazılım geliştirme sürecinin aşamalarıdır:

Gereksinim Analizi: Kullanıcı ihtiyaçları belirlenir.

Tasarım: Sistem ve arayüz tasarlanır.

Kodlama: Yazılım geliştirilir.

Test: Hatalar kontrol edilir.

Dağıtım: Yazılım yayınlanır.

Bakım: Güncellemeler ve iyileştirmeler yapılır.

## 15. Yapay Zeka ve Makine Öğrenmesi

Makine Öğrenmesi (Machine Learning - ML): Bilgisayarların veriyle öğrenmesini sağlar.

Derin Öğrenme (Deep Learning - DL): Sinir ağları kullanarak daha karmaşık problemlerin çözülmesini sağlar.

Veri Madenciliği: Büyük verilerden anlamlı bilgiler çıkarmaya yönelik işlemler.



## 16. Bilgisayar Mimarisi ve Donanım Temelleri

- Merkezi İşlem Birimi (CPU – Central Processing Unit):
  - Bilgisayarın beynidir, komutları işler.
  - Saat Hızı (Clock Speed): GHz cinsinden ölçülür, işlem hızını belirler.
  - Çekirdek Sayısı: Çok çekirdekli işlemciler daha fazla işlem yapabilir.
  - Önbellek (Cache): İşlemcinin hızlı erişim sağladığı bellek türüdür (L1, L2, L3).
- Bellek ve Depolama:
  - RAM (Random Access Memory): Geçici veri saklar, bilgisayar kapandığında sıfırlanır.
  - ROM (Read Only Memory): Kalıcıdır, sistemin temel yazılımını barındırır.
  - SSD (Solid State Drive): HDD'ye göre daha hızlıdır, hareketli parça içermez.
  - HDD (Hard Disk Drive): Manyetik disk yapısına sahip, veri saklama için kullanılır.
- Giriş ve Çıkış Birimleri:
  - Giriş Birimleri: Klavye, fare, mikrofon, tarayıcı.
  - Çıkış Birimleri: Ekran, yazıcı, hoparlör.

## 17. İşletim Sistemi (Operating System – OS)

- İşletim Sisteminin Görevleri:
  - Kaynak Yönetimi: Bellek, işlemci, disk gibi kaynakları yönetir.
  - Dosya Sistemi Yönetimi: Verileri organize eder.
  - Kullanıcı Arayüzü: Kullanıcının sistemle etkileşimini sağlar.
- Popüler İşletim Sistemleri:
  - Windows: Kullanıcı dostu, yaygın kullanılır.
  - Linux: Açık kaynak kodlu, güvenli.
  - MacOS: Apple cihazlarda kullanılır.
  - Android ve iOS: Mobil cihazlar için geliştirilmiştir.
- Sanal Bellek (Virtual Memory):
  - RAM yetersiz kaldığında, disk alanının RAM gibi kullanılmasını sağlar.
- Çoklu Görev (Multitasking):
  - Birden fazla işlemin aynı anda çalışmasını sağlar.

## 18. Programlama Temelleri ve Yazılım Dilleri

- Yüksek Seviye Programlama Dilleri:
  - Python: Basit ve öğrenmesi kolaydır, veri bilimi ve yapay zeka alanında kullanılır.
  - Java: Nesne yönelimli, taşınabilirlik açısından güçlüdür.
  - C++: Performans gerektiren uygulamalarda kullanılır.
  - JavaScript: Web geliştirme için kullanılır.
- Düşük Seviye Diller:
  - Assembly: Makine koduna çok yakın, donanım kontrolü için kullanılır.
  - Makine Dili (Binary Code – 0 ve 1'ler): En temel seviyedeki dil.
- Algoritma ve Veri Yapıları:
  - Diziler (Arrays): Sabit uzunlukta veri saklama yapılarıdır.
  - Bağlı Listeler (Linked Lists): Dinamik veri saklama yapılarıdır.
  - Ağaçlar (Trees): Hiyerarşik veri yapısıdır (Binary Tree, BST).
  - Yığın (Stack) ve Kuyruk (Queue): Veri ekleme ve çıkarma prensiplerine dayanır.

## 19. Yazılım Geliştirme Süreçleri ve Modelleri

- Yazılım Geliştirme Yaşam Döngüsü (SDLC – Software Development Life Cycle):
  - Gereksinim Analizi: Kullanıcının ihtiyacı belirlenir.
  - Tasarım: Mimari planlama yapılır.
  - Kodlama: Yazılım yazılır.
  - Test: Hatalar bulunur ve düzeltilir.
  - Dağıtım: Kullanıma sunulur.
  - Bakım: Güncellenir ve geliştirilir.
- Yazılım Geliştirme Modelleri:
  - Şelale Modeli (Waterfall Model): Adım adım ilerleyen geleneksel yöntem.
  - Çevik (Agile) Model: Esnek ve hızlı geri bildirim alan model.
  - Scrum: Takımların belirli periyotlarda (sprint) yazılım geliştirdiği Agile metodolojisi.





## 20. Yapay Zeka ve Veri Bilimi

- Yapay Zeka (Artificial Intelligence – AI):
  - İnsan gibi düşünen sistemler geliştirme alanıdır.
  - Makine Öğrenmesi (Machine Learning – ML): Veriden öğrenme yeteneği sağlar.
  - Derin Öğrenme (Deep Learning – DL): Sinir ağları kullanılarak daha karmaşık öğrenme işlemleri yapılır.
- Veri Bilimi ve Büyük Veri:
  - Büyük Veri (Big Data): Büyük ve karmaşık veri setlerinin analiz edilmesi.
  - Veri Madenciliği: Verilerden anlamlı bilgilerin çıkarılması.
  - SQL ve NoSQL:
    - SQL: Yapılandırılmış verileri yönetmek için kullanılır (MySQL, PostgreSQL).
    - NoSQL: Esnek ve büyük ölçekli verileri işler (MongoDB, Cassandra).

## 21. Siber Güvenlik ve Şifreleme

- Siber Güvenlik Tehditleri:
  - DDoS (Distributed Denial of Service): Ağ kaynaklarını çökertme saldırısı.
  - Phishing (Kimlik Avı): Sahte e-postalar ile kullanıcı bilgilerini çalma.
  - Brute Force Attack: Şifreleri deneme yanılma yöntemiyle kırma.
- Şifreleme Yöntemleri:
  - Simetrik Şifreleme: Aynı anahtar ile şifreleme ve çözme (AES, DES).
  - Asimetrik Şifreleme: Açık ve özel anahtar kullanır (RSA, ECC).
  - Hashing (Özetleme Algoritmaları): Verinin tek yönlü şifrenmesi (SHA, MD5).

## 22. Bulut Bilişim ve Dağıtık Sistemler

- Bulut Bilişim (Cloud Computing):
  - Veri ve uygulamaların internet üzerinden erişilmesini sağlar.
  - Hizmet Modelleri:
    - SaaS (Software as a Service): Google Drive, Dropbox gibi hizmetler.
    - PaaS (Platform as a Service): Google App Engine gibi geliştiricilere özel hizmetler.
    - IaaS (Infrastructure as a Service): Amazon Web Services (AWS), Microsoft Azure.
- Dağıtık Sistemler (Distributed Systems):
  - Verinin farklı bilgisayarlar arasında paylaşılmasını ve işlenmesini sağlar.
  - Örnek: Blockchain, merkezi olmayan bir veri yapısıdır.

## 23. IoT (Internet of Things – Nesnelerin İnterneti)

IoT Nedir?:

Cihazların internet üzerinden birbirine bağlanarak veri paylaşmasıdır.

Örneğin: Akıllı ev sistemleri, giyilebilir cihazlar (Apple Watch).

IoT'nin Bileşenleri:

Sensörler: Veri toplar.

Ağ Bağlantısı: Wi-Fi, Bluetooth ile iletişim kurar.

Veri İşleme: AI/ML algoritmaları kullanılarak analiz edilir.

## 18. Programlama Temelleri ve Yazılım Dilleri

- Yüksek Seviye Programlama Dilleri:
  - Python: Basit ve öğrenmesi kolaydır, veri bilimi ve yapay zeka alanında kullanılır.
  - Java: Nesne yönelimli, taşınabilirlik açısından güçlüdür.
  - C++: Performans gerektiren uygulamalarda kullanılır.
  - JavaScript: Web geliştirme için kullanılır.
- Düşük Seviye Diller:
  - Assembly: Makine koduna çok yakın, donanım kontrolü için kullanılır.
  - Makine Dili (Binary Code – 0 ve 1'ler): En temel seviyedeki dil.
- Algoritma ve Veri Yapıları:
  - Diziler (Arrays): Sabit uzunlukta veri saklama yapılarıdır.
  - Bağlı Listeler (Linked Lists): Dinamik veri saklama yapılarıdır.
  - Ağaçlar (Trees): Hiyerarşik veri yapısıdır (Binary Tree, BST).
  - Yığın (Stack) ve Kuyruk (Queue): Veri ekleme ve çıkarma prensiplerine dayanır.

## 19. Yazılım Geliştirme Süreçleri ve Modelleri

- Yazılım Geliştirme Yaşam Döngüsü (SDLC – Software Development Life Cycle):
  - Gereksinim Analizi: Kullanıcının ihtiyacı belirlenir.
  - Tasarım: Mimari planlama yapılır.
  - Kodlama: Yazılım yazılır.
  - Test: Hatalar bulunur ve düzeltilir.
  - Dağıtım: Kullanıma sunulur.
  - Bakım: Güncellenir ve geliştirilir.
- Yazılım Geliştirme Modelleri:
  - Şelale Modeli (Waterfall Model): Adım adım ilerleyen geleneksel yöntem.
  - Çevik (Agile) Model: Esnek ve hızlı geri bildirim alan model.
  - Scrum: Takımların belirli periyotlarda (sprint) yazılım geliştirdiği Agile metodolojisi.

## 1. Bilgisayar Mimarisi ve İşlemci (CPU)

- CPU (Central Processing Unit – Merkezi İşlem Birimi):
  - Aritmetik ve Mantık Birimi (ALU – Arithmetic Logic Unit): Matematiksel ve mantıksal işlemleri gerçekleştirir.
  - Kontrol Birimi (CU – Control Unit): Komutları yönetir ve sıralar.
  - Önbellek Belleği (Cache Memory): CPU’nun hızlı erişim sağladığı küçük, hızlı bellek alanıdır.
- Bellek (RAM – Random Access Memory):
  - Geçici veri saklama alanıdır.
  - Bilgisayar kapandığında veriler silinir.
- Makine Dili (Makine Kodu – Machine Code):
  - 1’ler ve 0’lardan oluşan en temel seviyedeki programlama dili.
  - İşlemciler doğrudan makine kodunu anlar.

## 2. Programlama Dilleri ve Gelişim Süreci

### Programlama dilleri 5 nesile ayrılır:

1. Nesil (1GL) – Makine Dili:
  - En düşük seviyeli dildir.
  - Kodlar direkt 1 ve 0 olarak yazılır.
2. Nesil (2GL) – Assembly Dili:
  - İnsan tarafından okunabilir ama hala düşük seviyeli bir dildir.
  - Assembler (çevirici) kullanılarak makine diline çevrilir.
3. Nesil (3GL) – Yüksek Seviyeli Programlama Dilleri:
  - C, C++, Python, Java, Pascal, Basic, Fortran, Cobol gibi diller.
  - Derleyiciler (compiler) kullanılarak makine diline çevrilir.
4. Nesil (4GL) – Veri Tabanı ve Raporlama Dilleri:
  - SQL, PL/SQL, MATLAB gibi diller.
  - Veri tabanı yönetimi ve hızlı uygulama geliştirme için kullanılır.
5. Nesil (5GL) – Yapay Zeka ve Mantıksal Programlama:
  - Prolog, Lisp gibi diller.
  - Yapay zeka, makine öğrenmesi ve uzman sistemlerde kullanılır.

## 3. Programlama Dillerinin Derlenmesi ve Çalıştırılması

- Derleme (Compile Time):
  - Kaynak kodun (C, Java, Python vb.) makine diline çevrilmesi işlemidir.
  - Derleyici (Compiler) kaynak kodu doğrudan makine koduna dönüştürür.
- Çalıştırma Zamanı (Run Time):
  - Programın çalıştırıldığı zamandır.
  - Yorumlayıcılar (Interpreter) her satırı anında çalıştırır (Python, JavaScript).
- Orta Seviye Diller (Intermediate Language):
  - C ve C++: Derlenen diller.
  - Java: Önce Bytecode’a derlenir, sonra JVM (Java Virtual Machine) tarafından çalıştırılır.
  - C#: IL (Intermediate Language) kullanır.



#### 4. Programlama Dillerinin Çalışma Mekanizması

- Derleyici (Compiler):
  - Kaynak kodu (C, C++, Java) tamamen makine koduna çevirir.
  - Çıktı .exe veya .bin gibi çalıştırılabilir dosyalardır.
  - Örnek: GCC (C/C++), Javac (Java).
- Yorumlayıcı (Interpreter):
  - Kaynak kodu satır satır çalıştırır.
  - Daha esnek ama daha yavaştır.
  - Örnek: Python, JavaScript, PHP.
- Hibrit Sistemler:
  - Java: İlk olarak Bytecode'a çevrilir, sonra JVM (Java Virtual Machine) tarafından çalıştırılır.
  - C#: İlk olarak IL (Intermediate Language)'e çevrilir, sonra .NET CLR (Common Language Runtime) tarafından çalıştırılır.

#### 5. Programlama Paradigmaları (Programlama Yaklaşımları)

- Prosedürel Programlama:
  - Kodlar fonksiyonlar (functions) halinde yazılır.
  - Örnek: C, Pascal.
- Nesne Yönelimli Programlama (Object-Oriented Programming – OOP):
  - Class (Sınıf) ve Object (Nesne) kullanarak kodlar organize edilir.
  - Kavramlar:
    - Kapsülleme (Encapsulation): Verilerin gizlenmesi.
    - Miras (Inheritance): Bir sınıfın başka bir sınıftan türemesi.
    - Çok Biçimlilik (Polymorphism): Aynı isimde farklı işlevlere sahip fonksiyonlar yazabilme.
  - Örnek: Java, C++, Python, C#.
- Fonksiyonel Programlama:
  - Fonksiyonlar birinci sınıf nesnelerdir.
  - Yan etkisiz (pure) fonksiyonlar kullanılır.
  - Örnek: Haskell, Lisp, Scala.
- Mantıksal Programlama:
  - Matematiksel mantık kurallarıyla çalışır.
  - Örnek: Prolog.

#### 6. Veritabanı Yönetimi ve SQL

- Veritabanı Nedir?
  - Yapılandırılmış veri depolamak için kullanılır.
  - SQL (Structured Query Language) kullanılarak yönetilir.

Veritabanı Türleri:

- İlişkisel Veritabanları (Relational Databases – RDBMS):
  - Tablolar (Tables), Satırlar (Rows), Sütunlar (Columns) içerir.
  - Örnek: MySQL, PostgreSQL, Oracle, MSSQL.

NoSQL Veritabanları:

- Esnek yapıya sahiptir, JSON veya belge tabanlıdır.
- Örnek: MongoDB, Firebase.

#### 7. Bellek Yönetimi

- Yığın (Stack) Belleği:
  - Küçük, hızlı ve LIFO (Last In, First Out) prensibiyle çalışır.
  - Fonksiyon çağrıları için kullanılır.
- Yığın (Heap) Belleği:
  - Büyük veriler için ayrılmış, dinamik bellektir.
  - Garbage Collector (Çöp toplayıcı) bellek yönetimi yapar (Java, Python).
- Bellek Sızıntısı (Memory Leak):
  - Belleğin gereksiz şekilde dolması.
  - C ve C++ gibi dillerde manuel bellek yönetimi gerekir.



- **8. Bilgisayar Ağları ve TCP/IP Modeli**

- TCP/IP Katmanları:

- Uygulama (Application) Katmanı: HTTP, FTP, DNS gibi protokoller burada çalışır.
- Taşıma (Transport) Katmanı: TCP ve UDP kullanarak veriyi taşır.
- İnternet (Internet) Katmanı: IP adresleme ve yönlendirme (Router).
- Ağ Erişim (Network Interface) Katmanı: Fiziksel kablolar ve bağlantılar.

- IP Adresleri:

- IPv4 (32-bit): 192.168.1.1 gibi.
- IPv6 (128-bit): Daha fazla cihaz destekler.

- Bağlantı Türleri:

- LAN (Local Area Network): Yerel ağ.
- WAN (Wide Area Network): Geniş alan ağı.
- VPN (Virtual Private Network): Güvenli ağ erişimi.

- **9. Siber Güvenlik Temelleri**

- Güvenlik Tehditleri:

- DDoS (Distributed Denial of Service): Ağı aşırı yükleyerek çökertme.
- Phishing: Kullanıcıları kandırarak bilgi çalma.
- MITM (Man in the Middle): Veri paketlerini izleme ve değiştirme.

- Şifreleme Yöntemleri:

- Simetrik Şifreleme: Aynı anahtar kullanılır (AES, DES).
- Asimetrik Şifreleme: Açık ve özel anahtar sistemi (RSA, ECC).

- **10. Yapay Zeka ve Makine Öğrenmesi**

- Yapay Zeka (AI – Artificial Intelligence):

- Bilgisayarların insan gibi karar vermesini sağlar.

- Makine Öğrenmesi (Machine Learning – ML):

- Gözetimli Öğrenme (Supervised Learning): Etiketlenmiş verilerle eğitim.
- Gözetimsiz Öğrenme (Unsupervised Learning): Verilerdeki desenleri keşfetme.

- Derin Öğrenme (Deep Learning):

- Sinir Ağları (Neural Networks) kullanarak büyük veri analizleri yapar.

- **11. Bulut Bilişim ve IoT**

- Bulut Bilişim (Cloud Computing):

- Verileri internet üzerinden erişilebilir hale getirir.
- Hizmet Modelleri:
  - SaaS (Software as a Service): Google Drive, Dropbox.
  - PaaS (Platform as a Service): Google App Engine.
  - IaaS (Infrastructure as a Service): Amazon Web Services (AWS).

- Nesnelerin İnterneti (IoT – Internet of Things):

- Cihazların internet üzerinden birbirine bağlanması.
- Örnek: Akıllı ev sistemleri, sağlık cihazları.



## • 1. Bilgisayarın Temel Gelişim Süreci

- John Napier (1550–1617):
  - Napier'in Kemikleri: Çarpma işlemini toplama problemine çevirmek için icat edilen bir sistem.
  - Logaritmik Sistem: Sürgü cetvelinin temelini oluşturdu.
- Blaise Pascal (1642):
  - Pascal'ın Toplama Makinesi: İlk mekanik hesap makinesi.
  - Dişli çarklarla çalışıyordu ve taşıma işlemi otomatik olarak gerçekleşiyordu.
- Gottfried Wilhelm Leibniz (1673):
  - Leibniz'in Hesap Makinesi: Çarpma, bölme ve karekök işlemlerini yapabiliyordu.
  - Dişli çarklar yerine silindir mekanizması kullanıldı.
- Marie Jacquard (1801):
  - Delikli Kartlı Tezgah (Jacquard Loom): Bilgi depolama ve otomatik veri işleme fikrini başlattı.
- 2. Charles Babbage ve Bilgisayarın Babası
- Fark Motoru (1822):
  - Logaritmalar ve polinomlar gibi matematiksel işlemleri hesaplayan ilk mekanik bilgisayar.
- Analitik Motor (1833):
  - İlk programlanabilir bilgisayar konsepti.
  - İlk Merkezi İşlem Birimi (CPU) ve Bellek Kavramı ortaya çıktı.
  - Programları delikli kartlara kaydetme fikri.
  - Lady Ada Lovelace, bu makinenin programlarını düzenlediği için ilk programcı kabul edilir.
- 3. Herman Hollerith ve İlk Veri İşleme Makinesi
- Hollerith'in Kart Okuyucusu (1877):
  - Delikli kartlarla veri saklama ve işleme fikrini geliştirdi.
  - ABD nüfus sayımı süresini 10 yıldan 3 yıla düşürdü.
  - IBM'in (International Business Machines) temellerini attı.

## • 4. İlk Bilgisayarlar ve Modern Bilgisayarların Doğuşu

- Mark I (1937–1943):
  - Harvard Üniversitesi ve IBM tarafından geliştirilen ilk elektromekanik bilgisayar.
  - İlk otomatik hesaplama yapan bilgisayar.
- ENIAC (1946):
  - İlk tamamen elektronik bilgisayar.
  - 18.000 vakum tüpü içeriyordu.
  - ABD ordusu tarafından kullanıldı (füze yörüngelerini hesaplamak için).
- EDVAC (1949):
  - İlk ikili sayı sistemini kullanan bilgisayar.
  - Von Neumann Mimarisi ile geliştirildi (Bellek içinde program saklama fikri).
- EDSAC (1949):
  - İlk program saklayabilen bilgisayar.
  - Döngü yapılarıyla program yazımı sağlandı.
- UNIVAC (1951):
  - İlk ticari bilgisayar.
  - General Electric tarafından kullanıldı.
  - IBM 650 (1955):
    - IBM'in bilgisayar sektöründeki liderliğini başlattı.



## • 1. Programlama Dillerinin Tarihçesi

- Kodlamanın tarihi 1840'lara kadar uzanır ve modern bilgisayar sistemlerinin temelini oluşturur.
- 1843 – Ada Lovelace:
  - İlk makine algoritmasını yarattı.
  - Charles Babbage'ın Analitik Motoru için bir program yazdı.
  - Bilgisayar bilimlerinin ilk programcısı olarak kabul edilir.
- 1936 – Alan Turing:
  - Evrensel Makine kavramını ortaya attı.
  - Modern bilgisayar biliminin temellerini attı.
  - Turing Makinesi, hesaplama mantığını modelleyen bir sistem olarak kabul edilir.
- 1940'lar – Konrad Zuse:
  - İlk programlama dili olarak kabul edilen Plankalkül'ü geliştirdi.
  - Kod depolama özelliği sayesinde tekrarlanan görevleri otomatikleştirdi.
- 2. Programlama Dillerinin Gelişim Süreci
  - Programlama dilleri, farklı evrim aşamalarından geçerek günümüzün gelişmiş dillerine ulaştı.

### 1949 – Assembly Dili ve Shortcode

- Assembly Dili:
  - Makine dili yerine insan tarafından okunabilir komutlar içerir.
  - Daha düşük seviyeli işlemler için kullanılır (Mikrodenetleyiciler, sistem programlama).
- Shortcode (Kısa Kod):
  - BINAC ve UNIVAC bilgisayarlarında kullanıldı.
  - Daha kolay ve daha hızlı programlama sağladı.
  - 1950'ler – Bilimsel Programlama Dillerinin Doğuşu
- 1952 – Autocode:
  - Makine koduna çevrilebilen ilk dildi.
  - Manchester Üniversitesi Mark 1 bilgisayarı için geliştirildi.
- 1957 – FORTRAN (Formula Translation):
  - Bilimsel ve matematiksel hesaplamalar için geliştirildi.
  - Günümüzde hâlâ süper bilgisayarlarda kullanılıyor.
- 1958 – ALGOL ve LISP:
  - ALGOL (Algorithmic Language): C, Pascal, Java, C++ gibi dillerin atasıdır.
  - LISP (List Processing): Yapay zeka (AI) ve veri işleme için tasarlandı.
- 1959 – COBOL (Common Business Oriented Language):
  - Dr. Grace Murray Hopper tarafından geliştirildi.
  - Bankacılık, sigortacılık ve ticari sistemlerde hâlâ kullanılıyor.

## • 3. 1960'lardan Sonra Programlama Dillerinin Yaygınlaşması

- Yüksek seviyeli diller, ticari ve akademik kullanımı artırdı.
- 1964 – BASIC (Beginners' All-purpose Symbolic Instruction Code):
  - Öğrenciler için geliştirildi.
  - Bill Gates ve Microsoft tarafından geliştirildi ve pazarlandı.
- 1970 – PASCAL:
  - Niklaus Wirth tarafından geliştirildi.
  - Kolay öğrenilebilir ve eğitim amaçlı kullanıldı.
- 1972 – C, Smalltalk ve SQL:
  - C (Dennis Ritchie): UNIX işletim sistemi için tasarlandı.
  - Smalltalk: Nesne yönelimli programlamanın temelini attı.
  - SQL (Structured Query Language): Veritabanı sorgulama ve yönetim dili.
- 4. 1980'lerde Nesne Yönelimli Programlama
  - Bu dönemde nesne yönelimli programlama ve veri yönetimi büyük gelişmeler kaydetti.
- 1980 – ADA:
  - Avrupa'daki hava trafik kontrolü ve uzay programlarında kullanıldı.
- 1983 – C++ ve Objective-C:
  - Bjarne Stroustrup tarafından geliştirilen C++, C'nin nesne yönelimli bir versiyonudur.
  - Objective-C, Apple'ın MacOS sistemlerinde kullanıldı.
- 1987 – Perl:
  - Metin düzenleme ve sistem yönetimi için geliştirilmiş genel amaçlı bir dildir.
- 5. 1990'larda Web ve Dinamik Diller
  - İnternetin yükselişiyle birlikte web geliştirme dillerine ihtiyaç arttı.
- 1991 – Python ve Visual Basic:
  - Python (Guido Van Rossum): Bugün en popüler dillerden biri.
  - Visual Basic: Windows tabanlı uygulamalar için geliştirildi.
- 1993 – Ruby:
  - Web geliştirme ve uygulama geliştirmede popüler bir dildir.
- 1995 – Java, JavaScript, PHP:
  - Java (James Gosling): Android ve birçok büyük sistemde kullanılır.
  - JavaScript (Brendan Eich): Web geliştirme için kullanılır.
  - PHP (Rasmus Lerdorf): Dinamik web sayfaları için geliştirilmiştir.



- **6. 21. Yüzyılda Programlama Dilleri**

- Modern programlama dilleri büyük ölçekli sistemler ve mobil uygulamalar için optimize edilmiştir.
- 2000 - C#:
- Microsoft tarafından geliştirildi.
- Visual Basic ve C'nin öğelerini birleştirir.
- 2003 - Scala ve Groovy:
- Scala, Java ile uyumludur ve büyük veri işleme için kullanılır.
- Groovy, kolay öğrenilebilir ve web uygulamalarında kullanılır.
- 2009 - Go (Google):
- Google tarafından büyük yazılım sistemleri için geliştirildi.
- 2014 - Swift (Apple):
- Apple tarafından iOS ve macOS uygulamalarını geliştirmek için oluşturuldu.

- **7. Günümüz ve Gelecekte Programlama**

- Günümüzde kullanılan popüler programlama dilleri ve trendler şunlardır:
- Python, JavaScript, C, C++, Java → En çok kullanılan diller.
- Yapay Zeka ve Veri Bilimi için Python ve R yaygın olarak kullanılıyor.
- Web geliştirme için JavaScript, TypeScript ve PHP tercih ediliyor.
- Mobil uygulamalar için Swift (iOS) ve Kotlin (Android) kullanılıyor.
- Büyük veri ve makine öğrenmesi için Python, Scala ve TensorFlow ön planda.
- Blok zinciri ve güvenlik için Solidity, Rust gibi diller popüler hale geliyor.



Viel Glück <3