GDI Übung 4

1.)
a) EVA = Eingabe Verarbeitung Ausgabe (Grundprinzip der Datenverarbeitung)

b)

Relaisrechner		Zuse Z3; Mark 1
Röhrenrechner	Trommelspeicher	ENIAC
Transistorrechner	Magnetkernspeicher	
IC-Rechner	IC-Schaltungen	
Mikrocomputer	Microprozessoren	

c)
Ein Speicher beinhaltet Daten und Programme an einem Ort.

d)
Hauptbestandteil: Prozessor besteht aus Steuerwerk und Rechenwerk

Arbeitsspeicher Speichert Daten und Programme an einem Ort

Bussystem für Kommunikation der Komponenten

Eingabe- Ausgabeschnittstellen

e)
Register: kleine schnelle Speicherbausteine die Operanden des Prozessors halten.
Arbeitsregister: speichert Operanden für die ALU
Befehlszählerregister: Adresse des nächsten auszuführenden Befehls
Befehlsregister: enthält aktuellen Befehl

mit einer Wortlänge von 32-Bit lassen sich 2^32-1 (4GB) Bytes adressieren.
Die letzten 4 GB können nicht adressiert werden.

f) schnelle Zwischenspeicher welche häufig genutzte Informationen für den Prozessor vorhalten.

g)
Prozessor-Register
Level 1 Cache
Level 2 Cache
Arbeitsspeicher / Festplatten-Cache-Speicher
Sekundärspeicher
Archivspeicher

2.)
a) Eine Rechenanlage mit Betriebssystem mit den Basisfähigkeiten Rechenfähigkeit, Speicherfähigkeit und Kooperationsfähigkeit.
b) Betriebssystem: Gesamtheit der Komponenten die dem Benutzer eine abstrakte Schnittstelle zur Hardware bieten.
c) Kernel des Betriebssystems liegt in einem Programm.
Mikro-Kernel ist modularisiert
d) Veredelung der Hardware
Verwaltung der Betriebsmittel

Steuerung und Kontrolle der Programmausführung
Schutzfunktionen
Anbieten von Diensten in Form von Schnittstellen
e) vervielfacht Interfaces zur Hardware so dass sie parallel nutzbar wird.
Scheinbare Parallelität wird mit Scheduling erreicht. Prozesse werden sequenziell ausgeführt.

3)
a) call-by-value: Funktionsargumente werden bei Aufruf kopiert und separat bearbeitet. Die Variablen des Aufrufers bleiben erhalten. Erhöht den Speicherbedarf bei großen Argumenten.
Call-by-reference: Funktionsargumente sind Referenzen auf Variablen des Aufrufers. Werden sie durch die aufgerufene Funktion verändert bleibt diese Veränderung nach der Rückkehr der Funktion erhalten. Komplexität des Programms steigt.