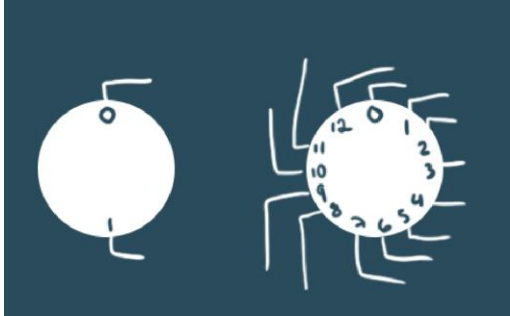


## Oppgave 1

**A)** Datamaskiner bruker binær logikk fordi det er enkelt å implementere, samt at når funksjonaliteten er bygget på et veldig enkelt konsept, blir systemet mer robust og mindre utsatt for feil. Dette gjør systemet rimeligere, samt øker ytelse og forutsigbarhet. Siden datamaskiner bruker strøm, men i varierende mengde, er det mer effektivt å benytte en "av/på"-switch enn å jobbe med større kalkulasjoner.



**B)**

(nivå, navn på lag, eksempler på protokoller)

5	Applikasjonslaget	HTTP, DNS, SMTP, FTP	DATA	Segment			
4	Transportlaget	TCP, UDP	DATA	TCP	Packet		
3	Nettverkslaget	Ruter, IP, ICMP, RIP	DATA	TCP	IP	Frame	
2	Datalinjelaget	Ethernet, IEEE, FDDI	DATA	TCP	IP	ETHERNET	
1	Fysisk	Kabler					

**Applikasjonslaget:** Oppgaven til applikasjonslaget er å definere en protokoll som inneholder informasjon om hvordan det skal leveres. Det handler kun om endesystemer

**Transportlaget:** Transportlaget oppretter kommunikasjon. Det gir en port, som gjør at meldingen fra applikasjonslaget havner i riktig applikasjon. Det kvalitetssikrer også kommunikasjonen mellom prosessene i en applikasjon. Protokollene til transportlaget varierer i pålitelighet. TCP lar deg ikke miste noe informasjon, da den kjører en «handshake» som bekrefter at alt som sendes blir sendt. UDP derimot, kvalitetssikrer ikke informasjonen på samme nivå. UDP er derimot mer effektivt, og egner seg for informasjon som kan tåle tap på veien – f.eks video og audiosamtaler, eller online spill.

**Nettverkslaget:** Jobber med å frakte en datapakke fra endesystem til endesystem fra kjernesystemet til nettverket.

**Datalinjelaget:** Overfører datapakkene mellom nabosystemer i nettverket

**Fysisk:** Overfører og koder hver enkelt bit gjennom et overføringsmedium

**C)** Et operativsystem (OS) er en abstraksjon av hardware, og gjør maskinen enklere å bruke av en bruker. Hovedoppgaven til et OS er å være brukervennlig, effektiv, og itererbar.

**Kernel Mode:** Kernel Mode lar brukeren bestemme helt fritt hvilke instruksjoner som skal kjøres på CPU'en. Brukeren har tilgang på alt minnet. Man vil heller ikke bli stoppet av OS'et hvis man prøver å gjøre endringer som kan være skadelige for maskinen.

**User Mode:** Når man kjører et program i user mode, vil det opprettes en prosess. Windows lar ikke prosesser bruke andre prosessers prosessorkraft, og vil dermed være limitert til designert plass. Man vil få advarsler, eller ingen mulighet, hvis man prøver å gjøre endringer som kan være skadelige eller påvirke maskinen på en negativ måte.

## Oppgave 2

**A)**

128	64	32	16	8	4	2	1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	1	1	1	0	1	1	1

$$495_{10} = 0000\ 0001\ 1110\ 1111_2$$

$$\begin{aligned} 495 - 256 &= 239 \\ 239 - 128 &= 111 \\ 111 - 64 &= 47 \\ 47 - 32 &= 15 \\ 15 - 8 &= 7 \\ 7 - 4 &= 3 \\ 3 - 2 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

128	64	32	16	8	4	2	1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	0	1	1	0	1	1	1

$$55_{10} = 0000\ 0000\ 0011\ 0111_2$$

$$\begin{aligned} 55 - 32 &= 23 \\ 23 - 16 &= 7 \\ 7 - 4 &= 3 \\ 3 - 2 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

**B)**

$$\begin{array}{r} \phantom{0}111\ 0101_2 \\ + 0010\ 0110_2 \\ \hline = 1001\ 1011_2 \end{array}$$

$$\begin{array}{r} \phantom{0}1011\ 0001_2 \\ + 1110\ 0100_2 \\ \hline = 1001\ 0101_2 \end{array}$$

C)

$$\begin{array}{r}
 301_{10} \\
 - 100_{10} \\
 \hline
 = 201_{10}
 \end{array}
 \quad
 \begin{array}{r}
 1\ 0010\ 1101_2 \\
 - 0\ 0110\ 0100_2 \\
 \hline
 = 0101\ 1001_2
 \end{array}$$

$0101\ 1001_2 = 201_{10}$

$0101\ 1001_2 + 1 = 0101\ 1010_2$

$0101\ 1010_2 = 202_{10}$

D) Unicode er et kodesett for elementer. Det inkluderer alle kjente tegn, samt emojis, fantasysspråk, og mer. Unicode er satt opp på 17 "plan", hvert plan består av 65536 kodepunkter. Unicode sier ikke noe om hvordan tegnene skal kodes binært, og man må derfor bruke et tegnkodeformat som f.eks UTF-8.

E)

$$U + 2655$$

$$0010\ 0110\ 0101\ 0101$$

$$1110\ 0010\ 1001\ 1001\ 1001\ 0101$$

F 2 9 9 9 5

$$0xE29995$$

F)

$$0x55 \& 0xF2 = 0x50$$

$$\begin{array}{r|l}
 0x55 & 0101\ 0101 \\
 \hline
 0xF2 & 1111\ 0010 \\
 \hline
 \& & 0101\ 0000
 \end{array}$$

$$0x39 \& 0xAC = 0x28$$

$$\begin{array}{r|l}
 0x39 & 0011\ 1001 \\
 \hline
 0xAC & 1010\ 1100 \\
 \hline
 \& & 0010\ 1000
 \end{array}$$

$$0x55 \mid 0x72 = 0x77$$

$$\begin{array}{r|l}
 0x55 & 0101\ 0101 \\
 \hline
 0x72 & 0111\ 0010 \\
 \hline
 \mid & 0111\ 0111
 \end{array}$$

$$0xFF \mid 0x34 = 0xFF$$

$$\begin{array}{r|l}
 0xFF & 1111\ 1111 \\
 \hline
 0x34 & 0011\ 0100 \\
 \hline
 \mid & 1111\ 1111
 \end{array}$$

$$0x69 \wedge 0xBB = 0xD2$$

$$\begin{array}{r|l}
 0x69 & 0110\ 1001 \\
 \hline
 0xBB & 1011\ 1011 \\
 \hline
 \wedge & 1101\ 0010
 \end{array}$$

G)

194 481	9261	441	21	1
$21^4$	$21^3$	$21^2$	$21^1$	$21^0$
	1	8	1	4

$$1814_{21} = \underline{\underline{12814_{10}}}$$

$$\begin{array}{r}
 1 \cdot 9261 \\
 + 8 \cdot 441 \\
 + 1 \cdot 21 \\
 + 4 \cdot 1 \\
 \hline
 = 12814
 \end{array}$$

### Oppgave 3

Oppgavene blir løst på Windows eller Mac. Det vil bli oppgitt i oppgaven hva som er benyttet. Jeg kommer til å fjerne navnet på brukeren min fra screenshots for å bevare anonymiteten.

#### A (Windows)

The screenshot shows a Windows Command Prompt window with the following text:

```

Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\>ping www.h-ck.me > list
  
```

Overlaid on the Command Prompt is a Notepad++ window titled "C:\Users\list - Notepad++". It contains the output of the ping command:

```

1
2 Pinging www.h-ck.me [46.30.215.126] with 32 bytes of data:
3 Reply from 46.30.215.126: bytes=32 time=10ms TTL=47
4 Reply from 46.30.215.126: bytes=32 time=13ms TTL=47
5 Reply from 46.30.215.126: bytes=32 time=10ms TTL=47
6 Reply from 46.30.215.126: bytes=32 time=10ms TTL=47
7
8 Ping statistics for 46.30.215.126:
9     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
10    Approximate round trip times in milli-seconds:
11        Minimum = 10ms, Maximum = 13ms, Average = 10ms
12
length: 470  Ln: 1  Col: 1  Pos: 1  Windows (CR LF)  UTF-8  INS
  
```

"Ping [www.h-ck.me](http://www.h-ck.me) > list" vil lagre resultatet av "ping [www.h-ck.me](http://www.h-ck.me)" som en tekstfil i filepathen du er inne i. Siden jeg kjører kommandoen fra C:\Users\XXX vil det opprettes en fil som heter "list" med ping-informasjonen fra h-ck.me.

**B (Windows)** For å se maskinens routing-table kan man bruke kommandoen **netstat -r**.  
Netstat vil gi deg statistikk for nettverkene dine, mens -r vil spesifikt gi deg routing-tabellen.

```

Command Prompt
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\>netstat -r

=====
Interface List
19...00 ff d8 c2 ed 6f .....TAP-Windows Adapter V9
9...04 d4 c4 4c 72 10 .....Intel(R) I211 Gigabit Network Connection
12...00 ff 73 6d 8f ea .....TAP-NordVPN Windows Adapter V9
13...56 a0 50 70 06 07 .....Microsoft Wi-Fi Direct Virtual Adapter
7...56 a0 50 70 0e 07 .....Microsoft Wi-Fi Direct Virtual Adapter #2
18...54 a0 50 70 06 07 .....Broadcom 802.11ac Network Adapter
22...5c f3 70 91 4b d7 .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
Network Destination        Netmask          Gateway           Interface         Metric
0.0.0.0                    0.0.0.0          192.168.68.1      192.168.68.122    25
127.0.0.0                  255.0.0.0        On-link           127.0.0.1         331
127.0.0.1                  255.255.255.255  On-link           127.0.0.1         331
127.255.255.255            255.255.255.255  On-link           127.0.0.1         331
192.168.68.0               255.255.255.0    On-link           192.168.68.122    281
192.168.68.122             255.255.255.255  On-link           192.168.68.122    281
192.168.68.255             255.255.255.255  On-link           192.168.68.122    281
224.0.0.0                  240.0.0.0        On-link           127.0.0.1         331
224.0.0.0                  240.0.0.0        On-link           192.168.68.122    281
255.255.255.255            255.255.255.255  On-link           127.0.0.1         331
255.255.255.255            255.255.255.255  On-link           192.168.68.122    281
=====
Persistent Routes:
None

IPv6 Route Table
=====
Active Routes:
If Metric Network Destination      Gateway
9        281 ::/0                  fe80::9ada:c4ff:fe3f:492c
1        331 ::1/128               On-link
9        281 fe80::/64              On-link
9        281 fe80::70d3:1ce3:8231:dac/128 On-link
1        331 ff00::/8                On-link
9        281 ff00::/8                On-link
=====
Persistent Routes:
None

```

## C (Windows)

**URL:** <http://www.eastwillsecurity.com/tk1104/oppgave3c.php?kandidatnr=10152>

**URL-syntax:** protokoll://host:port/path?query

**Innhold på nettsiden:** tekst med innholdet "Html response for oppgave 3c"

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 77.111.240.75

No.	Time	Source	Destination	Protocol	Length	Info
14	1.196170	192.168.68.122	77.111.240.75	TCP	66	51806 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
15	1.196251	192.168.68.122	77.111.240.75	TCP	66	51807 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
19	1.206046	77.111.240.75	192.168.68.122	TCP	62	80 → 51806 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 WS=128
20	1.206054	77.111.240.75	192.168.68.122	TCP	62	80 → 51807 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 WS=128
21	1.206093	192.168.68.122	77.111.240.75	TCP	54	51807 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
22	1.206093	192.168.68.122	77.111.240.75	TCP	54	51806 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
23	1.206206	192.168.68.122	77.111.240.75	HTTP	598	GET /tk1104/oppgave3c.php?kandidatnr=10152 HTTP/1.1
24	1.217954	77.111.240.75	192.168.68.122	TCP	60	80 → 51807 [ACK] Seq=1 Ack=545 Win=30336 Len=0
25	1.224545	77.111.240.75	192.168.68.122	HTTP	523	HTTP/1.1 200 OK (text/html)
35	1.263724	192.168.68.122	77.111.240.75	TCP	54	51807 → 80 [ACK] Seq=545 Ack=470 Win=262144 Len=0
88	6.216862	77.111.240.75	192.168.68.122	TCP	60	80 → 51806 [FIN, ACK] Seq=1 Ack=1 Win=29312 Len=0
89	6.216905	192.168.68.122	77.111.240.75	TCP	54	51806 → 80 [ACK] Seq=1 Ack=2 Win=262656 Len=0
90	6.223270	77.111.240.75	192.168.68.122	TCP	60	80 → 51807 [FIN, ACK] Seq=470 Ack=545 Win=30336 Len=0
91	6.223290	192.168.68.122	77.111.240.75	TCP	54	51807 → 80 [ACK] Seq=545 Ack=471 Win=262144 Len=0
198	11.439857	192.168.68.122	77.111.240.75	TCP	54	51806 → 80 [FIN, ACK] Seq=1 Ack=2 Win=262656 Len=0
199	11.439897	192.168.68.122	77.111.240.75	TCP	54	51807 → 80 [FIN, ACK] Seq=545 Ack=471 Win=262144 Len=0
208	11.451263	77.111.240.75	192.168.68.122	TCP	60	80 → 51807 [ACK] Seq=471 Ack=546 Win=30336 Len=0
210	11.451325	77.111.240.75	192.168.68.122	TCP	60	80 → 51806 [ACK] Seq=2 Ack=2 Win=29312 Len=0

Fra serveren får man en spørring og et svar fra HTTP.

### Spørring

```

v Hypertext Transfer Protocol
> GET /tk1104/oppgave3c.php?kandidatnr=10152 HTTP/1.1\r\n
Host: www.eastwillsecurity.com\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
Accept-Encoding: gzip, deflate\r\n
Accept-Language: nb-NO,nb;q=0.9,no;q=0.8,nn;q=0.7,en-US;q=0.6,en;q=0.5\r\n
\r\n
[Full request URI: http://www.eastwillsecurity.com/tk1104/oppgave3c.php?kandidatnr=10152]
[HTTP request 1/1]
[Response in frame: 25]

```

- **Rosa (GET):** Method/metode
- **Blå (/tk1104/oppgave3d...):** URL
- **Oransje (HTTP/1.1):** Versjon
- **Rød (A:B):** navn : verdi. Til sammen danner dette headerlinjene.
- **Gul (Full request ...):** Body

### Svar

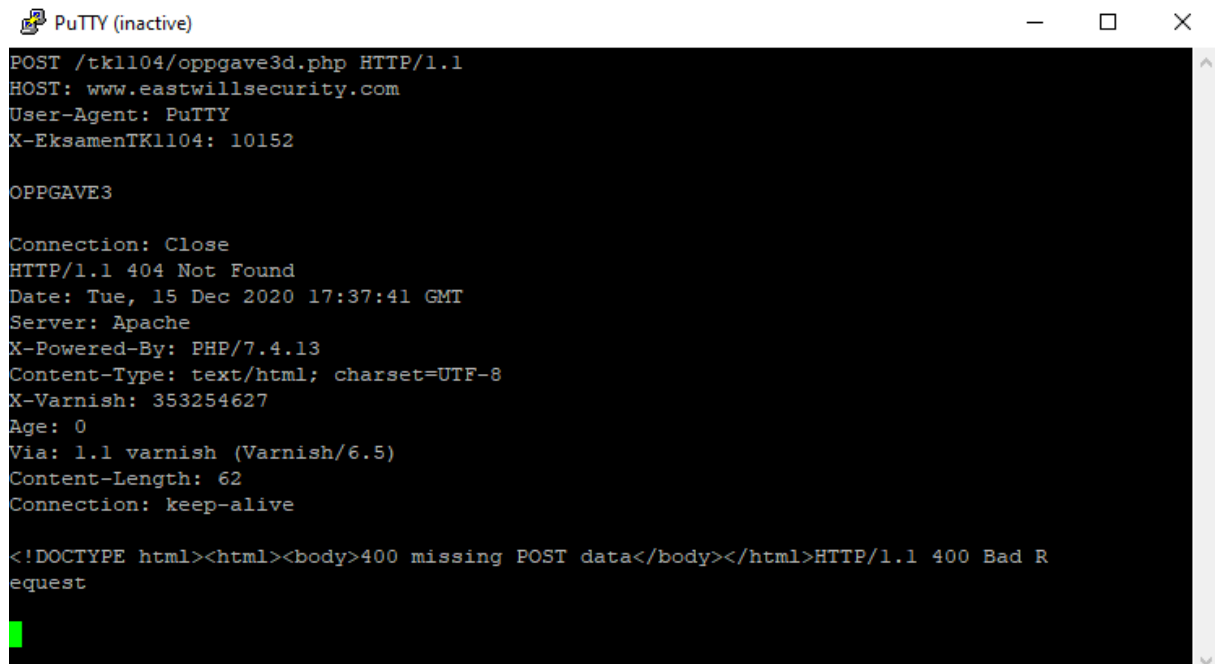
```

v Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
Date: Tue, 15 Dec 2020 11:51:20 GMT\r\n
Server: Apache\r\n
X-Powered-By: PHP/7.4.13\r\n
Expires: 0\r\n
Cache-Control: must-revalidate\r\n
X-Hash: 4720472040001015256176398\r\n
Vary: Accept-Encoding\r\n
Content-Encoding: gzip\r\n
Content-Length: 77\r\n
Content-Type: text/html;;charset=UTF-8\r\n
X-Varnish: 262341482\r\n
Age: 0\r\n
Via: 1.1 varnish (Varnish/6.5)\r\n
Accept-Ranges: bytes\r\n
Connection: keep-alive\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.018339000 seconds]
[Request in frame: 23]
[Request URI: http://www.eastwillsecurity.com/tk1104/oppgave3c.php?kandidatnr=10152]
Content-encoded entity body (gzip): 77 bytes -> 69 bytes
File Data: 69 bytes

```

- **Blå (HTTP/1.1):** Protokoll
- **Grønn (200 OK):** Statuskode
- **Rød (A:B):** navn : verdi. Til sammen danner dette headerlinjene.

## D (windows)



The screenshot shows a PuTTY terminal window titled "PuTTY (inactive)". The terminal displays the following text:

```
POST /tk1104/oppgave3d.php HTTP/1.1
HOST: www.eastwillsecurity.com
User-Agent: PuTTY
X-EksamenTK1104: 10152

OPPGAVE3

Connection: Close
HTTP/1.1 404 Not Found
Date: Tue, 15 Dec 2020 17:37:41 GMT
Server: Apache
X-Powered-By: PHP/7.4.13
Content-Type: text/html; charset=UTF-8
X-Varnish: 353254627
Age: 0
Via: 1.1 varnish (Varnish/6.5)
Content-Length: 62
Connection: keep-alive

<!DOCTYPE html><html><body>400 missing POST data</body></html>HTTP/1.1 400 Bad Request
```

### Forsøk:

*POST /tk1104/oppgave3d.php HTTP/1.1*  
*HOST: www.eastwillsecurity.com*  
*User-agent: PuTTY*  
*X-EksamenTK1104: 10152*

*OPPGAVE3*

*Connection: Close*

Jeg har prøvd å få til oppgaven en stund, men klarer ikke å få noe annet enn 404 og 400-errors.

Fremgangsmåte:

- **POST** (for å oppdatere innhold på siden under requesten)
- **/tk/1104/oppgave.3d.php** (pathen til siden man vil oppdatere)
- **http/1.1** (protokollen som benyttes)
- **HOST:** [www.eastwillsecurity.com](http://www.eastwillsecurity.com) (nettsiden som hoster pathen)
- **User-Agent: PuTTY** (Setter User-Agent-headeren til Putty)
- **X-EksamenTK1104: 10152** (Setter X-EksamenTK1104-headeren til kandidatnummeret)
- **Linjeskift – OPPGAVE3 – linjeskift** (Setter Body til «OPPGAVE3»)
- **Connection: Close** (lukker koblingen)

## Oppgave 4

**A)** Dette var i perioden da USA og Sovjet hadde et romkappløp, og Sovjet hadde allerede klart å sende Sputnik 2 i bane rundt jorda. Etter nederlaget grunna president Eisenhower Advanced Research Projects Agency Network (ARPA). ARPA var det første langutstrakte pakkelingsnettverket, som senere ble til det internettet vi kjenner i dag. Det startet som et forskningsprosjekt i 1966 grunnlagt av det Amerikanske forsvaret som et måte å koble til eksterne datamaskiner. De første datamaskinene ble koblet på nettverket i 1969, da fire universiteter ble koblet sammen (Utah, SRI, UCSB og UCLA). Selv om nettverket var på plass, så hadde man ikke satt opp felles protokoller. Dermed måtte alle lære hverandres protokollsetup for å kunne tolke dataen som ble sendt over. I 1970 kom den første ARPANET host-to-host protokollen, NPC (network control protocol). I desember 1970 hadde det originale nodesystemet til de fire skolene, blitt til 10 noder og 19 host-maskiner. Bare fire måneder senere var det 15 noder og 23 hosts. Det kom internasjonale noder i 1973 – da i England og Norge. Det kom også andre pakkesvitje-nettverk, utenfor ARPANET. ARPANET-prosjektet ble lagt ned i 1990, da det hadde kommet flere aktører på markedet og forskningsprosjektet ble ansett som fullført.

**B)**

Src port = 2114 <b>a</b>	Dest port = 53 <b>b</b>
Size = 12 <b>c</b>	Checksum = ???? ???? ???? ???? <b>d</b>
Data (binært) = 1100 1111 1111 0101 <b>e</b> 1100 1111 0001 0000	

**Handwritten calculations:**

```

a  0000 1000 0100 0010
b + 0000 0000 0011 0101
=  0000 1000 0111 0111

c + 0000 0000 0000 1100
=  0000 1000 1000 0011

d + 1100 1111 1111 0101
=  1101 1000 0111 1000

e + 1100 1111 0001 0000
=  1010 0111 1000 1000

+ 0001
=  1010 0111 1000 1001

```

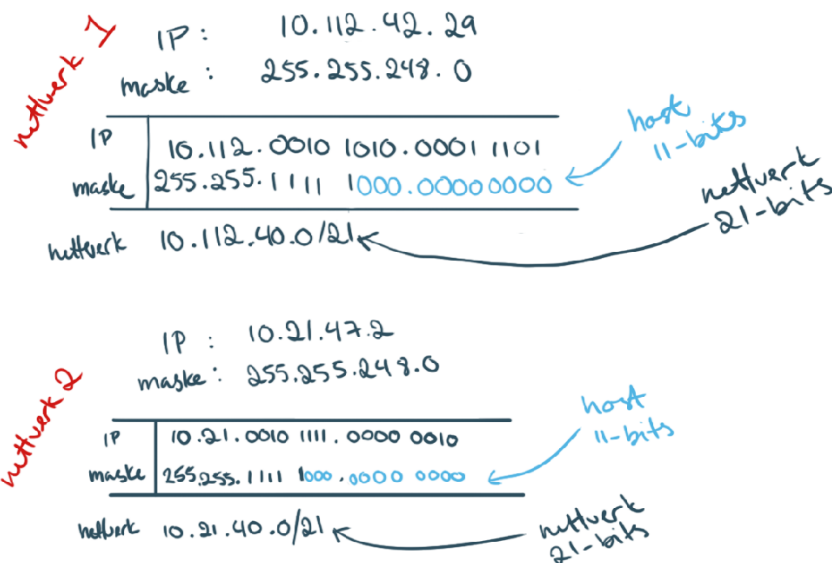
*checksum er enerkomplimentet, når man legger sammen sum + checksum skal svaret bare gi 1-ere*

checksum = 0101 1000 0111 0110

1. Gjør alle tall om til binær
2. Legg sammen portNr (a) og destPort (b)
3. Legg size til svaret du fikk fra a+b
4. Legg første 16-bit (til venstre, d) fra data til summen fra a+b+c
5. Legg siste 16-bit (til høyre, e) fra data til summen fra a+b+c+d
6. Hvis det blir overflow, legges den til i minst signifikante bit
7. For å finne checksummen, tar man enerkomplimentet til summen fra stegene over (a+b+c+d+e + eventuelle overflow)



c)



For at nettverk1 og nettverk2 skal være på samme subnet, så må tallene matche på alle verdier der masken er 255. Siden den første IP-adressen har 10.112 i IP-adressen der masken er 255, og nettverk2 har 10.21, kan vi se at de ikke er på samme subnet.

Selv om nettverkene ikke er på samme subnet, bør de kunne sende data til hverandre, da begge er på nettverk som kan sende og motta data. Dette blir ikke en direkte kobling.

**D) NAT** står for Network Address Translation. Nat brukes for å "forgrene" et, eller noen få, IP-adresser til flere enheter på samme nettverk. Internett trenger derfor bare å forholde seg til én IP-adresse, selv om det er flere koblet til. NAT vil gi egne adresser per enhet, og jobbe som en "oversetter" mellom IP'en internett har, og IP'en enheten på nettverket har. Dette gjør det mulig å legge til, fjerne, eller endre enheter internt uten å måtte informere omverden. Det gjør det også vanskelig for andre å vite nøyaktig hvilket device du er på, da man får oppgitt den offentlige IP'en, og ikke den personlige. Dette løser også et problem med at IP4-adresser kun har ca 4,3 milliarder unike adresser. Når hver person og bedrift har flere enheter, blir de fort brukt opp. Dette frigjør mye av behovet for IP4-behovet. Man har også begynt å innføre IP6, noe som løser deler av samme problem, men det er ikke universelt tatt i bruk, så man har ofte både en IP4 og en IP6-adresse.

