

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultetet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk og
informasjonsvitenskap

BOKMÅL



Sensurfrist: 7. januar 2008

Løsningsforslag: Avsluttende eksamen i TDT4110
Informasjonsteknologi, grunnkurs
Torsdag 6. desember 2007
9:00 – 13:00

Faglig kontakt under eksamen:

Alf Inge Wang, tlf 92289577

Hjelpemidler: C

Tilleggshefte I, "Introduksjon til HTML, CSS, JSP og MYSQL"

Typegodkjent kalkulator: HP30S

Sensur:

Resultater gjøres kjent på <http://studweb.ntnu.no> og sensurtelefon 81548014.

Notasjon:

Følgende notasjon for tabeller inklusive primær- og fremmednøkler er brukt i oppgaveteksten.
tabellnavn(primærnøkkel, vanligfelt, *fremmednøkkel*)

Oppgavesettet inneholder 5 oppgaver. Det er angitt i prosent hvor mye hver oppgave og hver deloppgave teller ved sensur. Les igjennom hele oppgavesettet før du begynner å lage løsning. Disponer tiden godt! Gjør rimelige antagelser der du mener oppgaveteksten er ufullstendig, skriv kort hva du antar.

Svar kort og klart, og skriv tydelig. Er svaret uklart eller lenger enn nødvendig trekker dette ned.

Lykke til!

Innhold:

- Oppgave 1: Flervalgsoppgave (20%)
- Oppgave 2: Grunnleggende programmering (30 %)
- Oppgave 3: JSP (30 %)
- Oppgave 4: HTML (10%)
- Oppgave 5: Databaser (10%)

Oppgave 1: Flervalgsoppgave (20%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på den ene selv). Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Kun ett svar er helt riktig. For hvert spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing eller mer enn ett kryss gir $-1/2$ poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne oppgaven. Der det er spesielle uttrykk står den engelske oversettelsen i parentes.

1. Brunfrid har klart å knekke passordet til en database med personalinformasjon på NTNU etter flere forsøk, men får kalde føtter og bestemmer seg for ikke å få innsyn i databasen. Har Brunfrid gjort noe ulovlig i følge loven?
 - a. Nei
 - b. Ja
 - c. Loven sier ikke noe om dette
2. Hva er hvit-boks testing i systemutvikling?
 - a. Testing som innebærer at man tar i betraktning systemets eller modulens indre oppbygning
 - b. Testing som innbærer at man ikke ser på systemets eller modulens indre oppbygning
 - c. Testing hvor man ser hvordan de ulike delene i et system fungerer sammen
3. Hvor mange bit trenger du til å kode et symbol i alfabetet $\Omega=\{A,B,C,D\}$ ved enkel koding hvor alle symbolene er representert med samme antall bit.
 - a. 2 bits
 - b. 3 bits
 - c. 4 bits
4. Hvilken type minne har normalt lavest tilgangstid (access time) i en datamaskin?
 - a. Primærminne
 - b. Cache
 - c. Registre
5. Man må søke Datatilsynet om konsesjon:
 - a. Man trenger aldri å søke Datatilsynet om konsesjon
 - b. Når man oppretter et personregister med sensitive personopplysninger
 - c. Alltid når man oppretter alle typer personregister
6. En ulempe med åpen kildekode (open source) er:
 - a. Det er vanligvis dyrt å anskaffe
 - b. Du har ingen mulighet til å endre på produktet selv
 - c. Du har normalt ikke krav på brukerstøtte
7. En ruter (router):
 - a. Er kun en signalforsterker
 - b. Den ser fysiske adresser og kan dermed kun splitte et nettverk
 - c. Den ser nettverksadresser og kan derfor koble sammen forskjellige typer nettverk

8. En vanlig datamaskin (PC) inneholder typisk:
 - a. Prosessor, primærlager, I/O-enheter og systembuss
 - b. Prosessor, primærlager, DNS-lager, I/O-enheter og systembuss
 - c. Prosessor, primærlager, I/O-enheter og GSM-bro
9. En prosessor:
 - a. Er flink til å legge sammen to tall
 - b. Får raskt oversikt over store bilder
 - c. Er flink til å løse abstrakte problemer uten menneskelig inngripen
10. Når er binærsøk normalt raskere enn sekvensielt søk?
 - a. Binærsøk er alltid raskere enn sekvensielt søk
 - b. Binærsøk er aldri raskere enn sekvensielt søk
 - c. Binærsøk fungerer bare på sorterte datamengder og er her normalt raskere enn sekvensielt søk
11. Hva er hovedoppgaven til en navnetjener?
 - a) Oversetter IP-adresser til domenenavn
 - b) Oversetter domenenavn til IP-adresser
 - c) Holder rede på variabelnavn når et program kjører
12. En protokoll er:
 - a. Et sett av kommunikasjonsregler
 - b. Et program som oversetter kildekode til kode som kan kjøres på en datamaskin
 - c. Hovedkomponent i en vanlig datamaskin
13. I det binære tallsystemet, hva er summen av 010 og 101?
 - a. 101
 - b. 1001
 - c. 111
14. I det binære tallsystemet, hva er 010 multiplisert med 101?
 - a. 111
 - b. 1010
 - c. 1111
15. Hva er inspeksjon i systemutvikling?
 - a. Inspeksjon innebærer å finne feil eller forsikre seg om at feil ikke finnes uten å kjøre programmet
 - b. Inspeksjon innebærer å undersøke om de som programmerer lager ryddige programmer som er lette å lese
 - c. Inspeksjon innebærer å finne feil i koden ved å kjøre programmet
16. Hvilket av følgende utsagn er *usant*?
 - a. Protokollen HTTP befinner seg i applikasjonslaget og bygger på TCP/IP
 - b. Nettverkslaget har ansvaret for å rute (videresende) pakkene i nettet
 - c. En melding som skal sendes over Internett deles opp i pakker. Pakkene kobles så sammen og sendes sammen over nettet

17. Gitt ei sortert liste av 280 verdier. Vi søker etter en verdi som ikke finnes i lista. Hvor mange sammenligninger må vi i verste fall gjøre ved bruk av binærsøk?
- 140
 - 8
 - 9
18. Hva er usant om adresser på Internett?
- En IP-adresse kan representeres vha det vanlige tallsystemet som 4 tall fra og med 0 og til og med 255 atskilt med punktum
 - En IP-adresse er på 32 bit, noe som betyr at det finnes 2^{32} unike IP-adresser
 - Et system som heter DAS (Domain Access System) holder orden på koblingen mellom en tekstlig adresse (for eksempel www.vg.no) og den tilhørende IP-adressen
19. Gitt at vi skal søke etter tallet 4 i den sorterte datamengden 1,4,8,9,11,14,23. Hvilke tall vil bli sjekket dersom vi anvender algoritmen binærsøk (ved partall antall elementer rund nedover)?
- 1,4
 - 9,4
 - 9,1,4
20. Hva defineres ikke som sensitiv informasjon iflg. Personopplysningsloven?
- Rasemessig eller etnisk bakgrunn, eller politisk, filosofisk eller religiøs oppfatning
 - Eksamensresultater
 - Medlemskap i fagforeninger

Løsningsforslag oppgave 1:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
b	a	a	c	b	C	c	a	a	c	b	a	c	b	a	c	c	c	b	b

Oppgave 2 – Grunnleggende programmering (30 %)

a) (5 %) Skriv en metode *hilb* som tar inn et heltall n og returnerer en tabell av flyttall (desimaltall) med n rader og n kolonner hvor hvert element har verdien $H(i, j) = 1/(i+j+1)$.

b) (10%) Lag metoden *kodMelding* som tar inn en tabell av like antall heltall og returnerer en kodet versjon av tabellen hvor to og to elementer parvis har byttet plass. Hvis metoden får inn $L_1, L_2, L_3, L_4, \dots$ skal $L_2, L_1, L_4, L_3, \dots$ returneres. Eksempel: med lista 12,65,34,76 skal metoden returnere 65,12,76,34.

c) (5%) Gjør nødvendige endringer for at denne metoden skal sortere lista i *synkende* i stedet for stigende rekkefølge. Skriv ned både linjenummer og ny programkode.

1	<%!
2	int[] sorter(int[] tabell) {
3	boolean harByttet = true ;
4	int temp;
5	while (harByttet) {
6	harByttet = false ;
7	for (int i=1; i<tabell.length;i++) {
8	if (tabell[i]<tabell[i-1]) {
9	temp = tabell[i];
10	tabell[i] = tabell[i-1];
11	tabell[i-1] = temp;
12	harByttet = true ;
13	}
14	}
15	}
16	return tabell;
17	}
18	%>

d) (10 %) Fullfør programmet `registrer.jsp` (vist under) som mottar parameteren `innhold` fra et webskjema og skal registrere innholdet i en julegave i en database samt antallet julegaver av samme type. Du skal ikke å gjenta koden som er oppgitt for `registrer.jsp` i besvarelsen. Programmet har følgende krav:

1. Databasen består av en tabell: `gave(artikkel, antall)`
2. Hvis innholdet av gaven ikke finnes fra før, skal man registrere innholdet og at det finnes 1 stk.
3. Hvis innholdet av gaven finnes fra før, skal man kun oppdatere antallet av denne gaven (artikkelen).
4. Programmet skal skrive følgende ut til brukeren (eks. `innhold=sokker`):

”Registrert gave: sokker. Har antall: 1 stk.”

`registrer.jsp`:

1	<code><%@page import="java.sql.*" %></code>
2	<code><%</code>
3	<code>Class.forName("org.gjt.mm.mysql.Driver").newInstance();</code>
4	<code>Connection db = DriverManager.getConnection(</code>
5	<code>"jdbc:mysql://mysql.stud.ntnu.no/alfw_itgk",</code>
6	<code>"alfw_itgkbrk", "passord");</code>
7	<code>Statement st = db.createStatement();</code>
8	<code>ResultSet rs;</code>
9	<code>String innhold = request.getParameter("innhold");</code>
10	<code>int antall=0;</code>

Løsningsforslag oppgave 2:**2a)**

```

<%!
double[][] hilb(int n) {
    double[][] tabell = new double[n][n];
    for (int i=0;i<n;i++) {
        for(int j=0;j<n;j++) {
            tabell[i][j] = 1.0/(i+j+1);
        }
    }
    return tabell;
}
%>

```

2b)

```

<%!
int[] kodMelding(int[] melding) {
    int temp;
    for (int i=1;i<melding.length;i=i+2) {
        temp=melding[i];
        melding[i]=melding[i-1];
        melding[i-1]=temp;
    }
    return melding;
}
%>

```

2c)

```

8:    if (tabell[i]>tabell[i-1]) {

```

2d)

```

rs = st.executeQuery("SELECT antall FROM gave
    WHERE artikkel='"+innhold+"'");
if (rs.next()) {
    antall = rs.getInt(1);
    st.executeUpdate("UPDATE gave SET antall=antall+1 WHERE
        artikkel='"+innhold+"'");
} else {
    st.executeUpdate("INSERT INTO gave (artikkel,antall)
        VALUES ('"+innhold+",1)");
}
antall++;
out.println("Registrert gave: "+innhold+". Har antall: "+antall+" stk.");
db.close();
%>

```


Oppgave 3 – JSP (30 %)

Eva har skrevet metoden bubbleSort i IT-grunnkurs. På Internett finner hun pseudokoden til en variant av boblesortering, metoden combSort:

```

INPUT liste L av tilfeldige heltall
OUTPUT sortert liste L
INITIALLY swaps = false and gap = size of L
while gap > 1 or swaps have been done
  if gap > 1
    gap = heltallsdelen av gap / 1.3
    if gap is 10 or 9, set gap = 11
  end if gap
  swaps = false
  foreach Li from first element until i+gap > last element
    if Li > Li+gap
      swap elements Li and Li+gap
      swaps = true
    end if swap
  end foreach
end while

```

a) (5 %) Eva er ikke helt sikker på om pseudokoden er riktig hvis lista L består av bare ett tall. Er den det? (begrunn svaret med maks 2 setninger)

b) (15 %) Anta pseudokoden er riktig uansett lengde på lista og skriv metoden combSort fra oppgitt pseudokode. Bruk følgende for å finn heltallsdelen av gap/1.3:

```
gap = (int)((double) gap/1.3);
```

c) (10 %) Eva lurer på hva som kjører raskest av bubbleSort og combSort. For å teste dette bruker hun metodene `currentTimeMillis` og `Math.random` som er innebygd i JSP. Metodene er beskrevet som følgende:

currentTimeMillis Returns the current time in milliseconds.

Eksempel på bruk: `long tid = System.currentTimeMillis();`

`long` er et heltall i JSP representert med 64bit, mens `int` er heltall representert med 32bit.

Math.random Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

Eksempel på bruk: `int tilfeldigHeltall = (int) (Math.random()*10);`

Eva sjekker ut dette ved å lage ei stor liste av tilfeldige heltall. Deretter sorterer hun lista med hver metode og måler tiden med `currentTimeMillis`. Til slutt skriver hun ut kjøretiden (hvor mange millisekunder metoden bruker) til bubbleSort samt kjøretiden til combSort i nettleseren. Lag et program som gjør dette. Programmet skal kun kalle metodene bubbleSort og combSort (skal ikke å skrive kode for bubbleSort).

Løsningsforslag oppgave 3:**3a)**

Ja. Hvis initialiseringen `swap = false` fjernes vil den feile for lengde 1 men fungere korrekt ellers.

3b) Etter som liste er et objekt, trenger man ikke å returnere lista.

```
<%!
combSort(int[] liste) {
    boolean swaps=false;
    int gap=liste.length;
    while (gap>1 || swaps) {
        if (gap>1) {
            gap = (int)((double) gap/1.3);
            if (gap==10 || gap==9) {
                gap=11;
            }
        } // End gap
        swaps=false;
        for (int i=0;i<liste.length-gap;i++) {
            if (liste[i]>liste[i+gap]) {
                int temp=liste[i];
                liste[i]=liste[i+gap];
                liste[i+gap]=temp;
                swaps=true;
            } // End swap
        } // End for
    } // End while
}
%>
```

3c)

```
int N=100000;
int[] bubbleListe = new int[N];
int[] combListe = new int[N];

for (int i=0;i<N;i++) {
    bubbleListe[i]=(int) (Math.random()*N);
    combListe[i]=(int) (Math.random()*N);
}

long t0=System.currentTimeMillis();
bubbleSort(bubbleListe);
long tb=System.currentTimeMillis()-t0;

t0=System.currentTimeMillis();
combSort(combListe);
long tc=System.currentTimeMillis()-t0;
%>
Tidsbruk BobbleSort: <%= tb %> <br/>
Tidsbruk CombSort: <%= tc %>
```

Oppgave 4: HTML (10%)

I denne oppgaven får du oppgitt en html-fil med 10 mangler. Webskjemaet skal bruke metoden "get" og innholdet skal sendes til programmet "registrer.cgi". Merk følgende krav:

1. Overskrifter av type <h3> skal være i *store bokstaver* og ha *fargen rød*.
2. Overskrifte av type <h4> skal ha *fontstørrelse 12pt*, ha *fonten courier* og være *grønn*.
3. Tabellen har strektykkelse 1 og det skal være *10 punkts avstand* mellom innholdet i en celle i tabellen (f.eks. tekst) og kanten av cellen.

Du skal kun skrive HTML-kode for de tomme feltene slik at nettleseren vil vise koden slik som vist i skjermbildet under:

Football team:	<input type="text" value="Rosenborg"/>	Description:	<input type="text" value="Great!"/>
THANKS FOR SUBMITTING...		<input type="button" value="Reset"/>	<input type="button" value="Submit"/>

Now: For det meste skyet, 3° C Tor: 3° C Fre: 0° C Lør: 1° C Søn:

(oppgaven fortsetter på neste side)

Html-filen

1	<html>
2	<head>
3	<title>Football Webpage</title>
4	<style type="text/css">
5	Fyll inn 1
6	Fyll inn 2
7	</style>
8	</head>
9	<body>
10	Fyll inn 3
11	Fyll inn 4
12	<tr>
13	<td>
14	<h4>Football team:</h4>
15	</td>
16	<td>
17	Fyll inn 5
18	</td>
19	Fyll inn 6
20	<td>
21	<select name="description">
22	<option>Great!</option>
23	</select>
24	</td>
25	</tr>
26	<tr>
27	Fyll inn 7
28	<h3>Thanks for submitting...</h3>
29	</td>
30	Fyll inn 8
31	Fyll inn 9
32	Fyll inn 10
33	</td>
34	</tr>
35	</table>
36	</form>
37	</body>
38	</html>

Løsning oppgave 4:

Html-filen

1	<html>
2	<head>
3	<title>Football Webpage</title>
4	<style type="text/css">
5	h3{text-transform:uppercase; color:red}
6	h4{font-size:12pt; font-family:courier; color:green}
7	</style>
8	</head>
9	<body>
10	<form action="registrer.cgi" type="get">
11	<table border="1" cellpadding="10">
12	<tr>
13	<td>
14	<h4>Football team:</h4>
15	</td>
16	<td>
17	<input type="text" name="team" value="Rosenborg"/>
18	</td>
19	<th>Description:</th>
20	<td>
21	<select name="description">
22	<option>Great!</option>
23	</select>
24	</td>
25	</tr>
26	<tr>
27	<td colspan="2">
28	<h3>Thanks for submitting...</h3>
29	</td>
30	<td colspan="2">
31	<input type="reset" value="Reset">
32	<input type="submit" value="Submit">
33	</td>
34	</tr>
35	</table>
36	</form>
37	</body>
38	</html>

Oppgave 5 : Databaser (10%)

Vi betrakter en bedrift som produserer produkter. For å lage et produkt trenger en forskjellige materialer. Materialene kjøpes hos leverandører. Et produkt identifiseres ved hjelp av en produktkode (**prodID**). I tillegg har produktet et navn (**prodNavn**) og en pris (**prodPris**). Et material identifiseres ved en kode (**matID**). I tillegg har materialet et navn (**matNavn**) og en enhetspris (**enhetsPris**). En leverandør av materialer identifiseres ved et leverandørnavn (**levNavn**). I tillegg registreres leverandørens adresse (**levAdresse**). Bedriften ønsker å registrere antall enheter (**antEnheter**) av hvert material som inngår i et produkt.

Vi har følgende business rules / beskrankninger :

1. Et produkt inneholder mange materialer
2. Alle produkter inneholder materialer
3. Samme material kan inngå i mange produkter
4. Alle materialer inngår i produkter
5. En leverandør kan kun levere ett material
6. Et material leveres kun av en leverandør
7. Alle leverandører leverer materialer

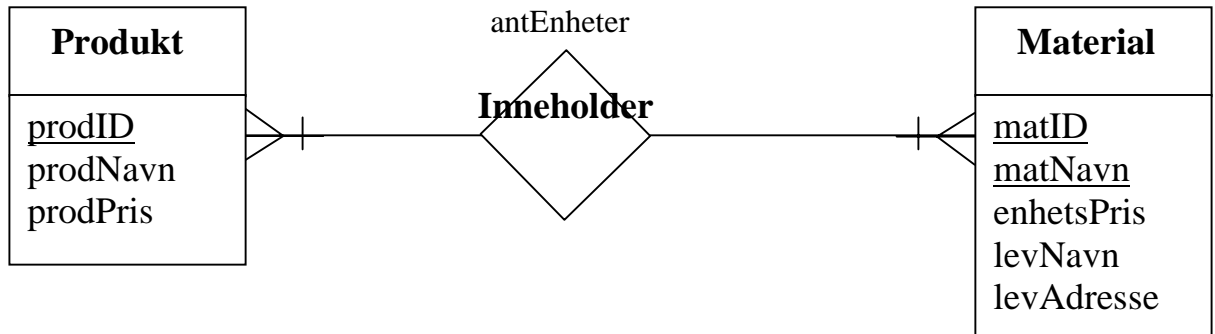
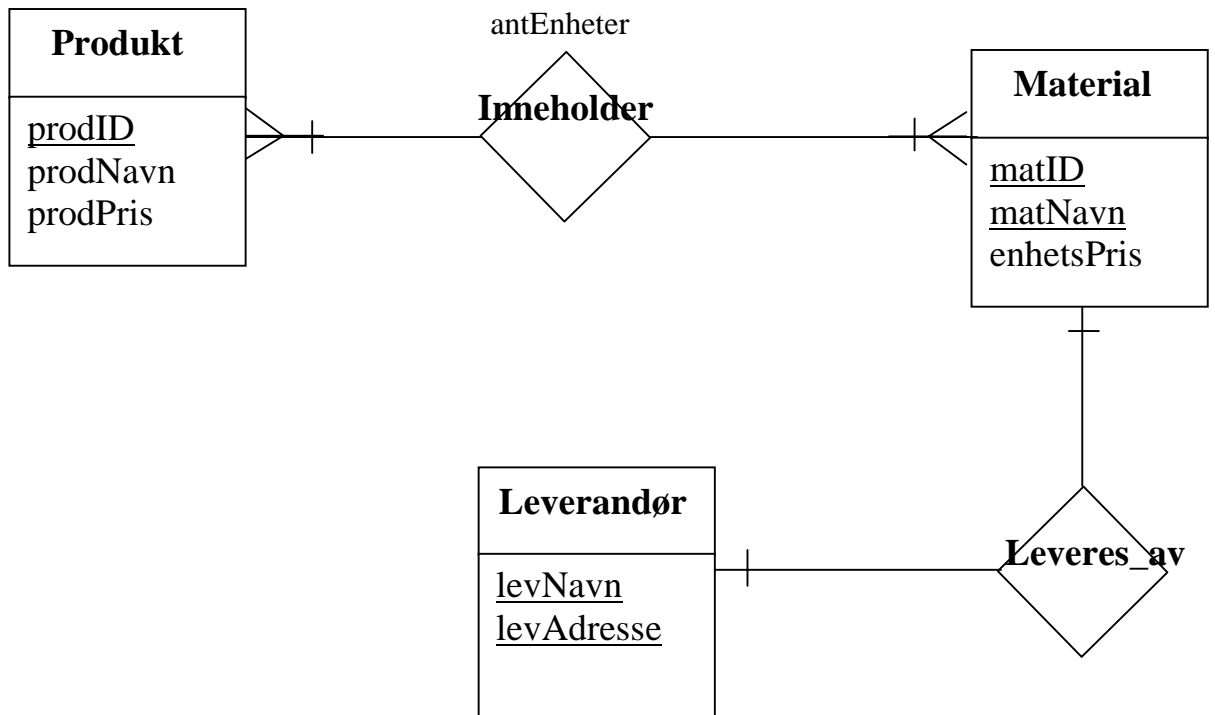
a) (4%) Lag et ER – diagram som modellerer en database basert på opplysningene først i oppgaven (en database som viser forholdet mellom produkter, materialer og materialleverandører).

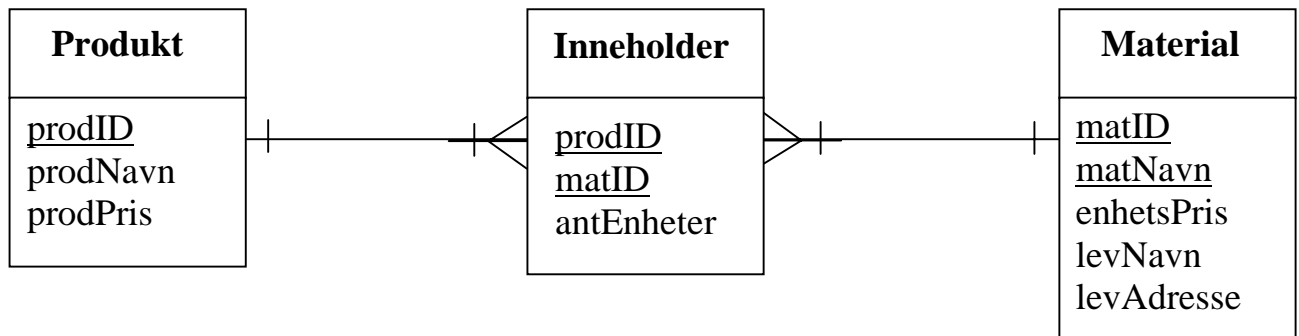
b) (3%) Skriv ned SQL-instruksjonene som er nødvendig for å opprette databasen som er modellert under **a)**. Velg selv fornuftige datatyper for attributtene.

c) (3%) Lag et query i SQL som produserer en oversikt som viser hvilke leverandører, hjemmehørende i Bergen, som har levert materialer til de forskjellige produktene (listen skal inneholde produktnavn og navn på materialleverandør). Oversikten skal sorteres på **prodID**.

Løsningsforslag oppgave 5:

a)

**ELLER :**

Alternativ utførelse (entetisering) :

b)

CREATE TABLE Produkt

```

(prodID      int PRIMARY KEY,
 prodNavn    varchar(20) NOT NULL,
 prodPris    int);
  
```

CREATE TABLE Material

```

(matID      int PRIMARY KEY,
 matNavn    varchar(20) NOT NULL,
 enhetsPris int,
 levNavn    varchar (20),
 levAdresse varchar(30));
  
```

CREATE TABLE Innold

```

(prodID      int,
 matID       int,
 antEnheter  int,
 PRIMARY KEY(prodID, matID));
  
```

c)

```

SELECT      prodNavn, levNavn
FROM        Produkt, Innhold, Material
WHERE       Produkt.prodID = Innhold.prodID AND
           Material.matID = Innhold.matID AND
           levAdresse = 'BERGEN'

ORDER BY   Produkt.prodID;
  
```


Svarskjema flervalgsoppgave (LØSNING)

Studentnummer: _____ Linje: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

<i>Oppgavenr</i>	<i>A</i>	<i>B</i>	<i>C</i>
1.1		O	
1.2	O		
1.3	O		
1.4			O
1.5		O	
1.6			O
1.7			O
1.8	O		
1.9	O		
1.10			O
1.11		O	
1.12	O		
1.13			O
1.14		O	
1.15	O		
1.16			O
1.17			O
1.18			O
1.19		O	
1.20		O	