

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultetet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk og
informasjonsvitenskap

BOKMÅL



Sensurfrist: xxx 2010

Avsluttende eksamen i TDT4110/IT1102
Informasjonsteknologi, grunnkurs
Snorsdag x. august 2010
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 4 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: Mer programmering (40 %)
- Oppgave 4: HTML (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. Hvor mange bit trenger du til å kode det norske alfabetet (29 forskjellige bokstaver) ved enkel koding hvor alle symbolene er representert med samme antall bit.
 - a. **5 bits**
 - b. 6 bits
 - c. 8 bits
2. Kalle har klart å knekke passordet til en database som lagrer alle karakterene på NTNU etter flere forsøk, men ombestemmer seg og ser ikke på innholdet i databasen. Har Kalle gjort noe ulovlig i følge loven?
 - a. Nei
 - b. **Ja**
 - c. Loven sier ikke noe om dette
3. Hvilken type minne har normalt høyest tilgangstid (access time) i en datamaskin?
 - a. Register
 - b. Cache
 - c. **Primærminne**
4. En ulempe med åpen kildekode (open source) er:
 - a. Du kan ikke endre produktet selv
 - b. Det er store firma står som regel bak utviklingen
 - c. **Du har normalt ikke krav på brukerstøtte**
5. Hva er beskriver en kravspesifikasjon?
 - a. **Kravspesifikasjon beskriver HVA systemet skal gjøre**
 - b. Kravspesifikasjon beskriver HVORDAN systemet skal lages
 - c. Kravspesifikasjon beskriver koden til et system
6. Hva er svart-boks testing?
 - a. Testing som konsentrerer seg om systemets eller modulens indre oppbygning
 - b. **Testing som ikke ser på systemets eller modulens indre oppbygning**
 - c. Testing som konsentrerer seg om hvordan de ulike delene i et system fungerer sammen
7. Hva er kjernen i et operativsystem?
 - a. Den delen som tar i mot input fra brukeren
 - b. **Den delen som administrerer maskinressurser**
 - c. Det mediet som operativsystemet installeres fra
8. Hvorfor krypteres data?
 - a. **For at data ikke skal kunne leses av uvedkommende**
 - b. For at det skal ta mindre plass
 - c. For at det skal kunne gjøres tilgjengelig for alle på Internett
9. En protokoll er:
 - a. Et program som oversetter kildekode til maskinkode slik at det kan kjøres på en datamaskin
 - b. En enhet som gjør beregninger og sammenlikninger i en datamaskin
 - c. **Et sett av kommunikasjonsregler**

10. Er binærsøk alltid raskere enn sekvensielt søk?
- Binærsøk er alltid raskere enn sekvensielt søk uansett
 - Binærsøk er alltid raskere enn sekvensielt søk på sorterte datamengder
 - Sekvensielt søk kan være raskere enn binærsøk**
11. Hva gjør man når man designer et system under systemutvikling?
- Beskriver hva systemet skal gjøre
 - Beskriver hvordan systemet skal lages**
 - Beskriver kun hvordan systemet skal se ut
12. Hva er hovedoppgaven til en navnetjener?
- Oversetter domenenavn til IP-adresser**
 - Oversetter IP-adresser til domenenavn
 - Oversetter IP-adresser til URL
13. I det binære tallsystemet, hva er 010 multiplisert med 101?
- 111
 - 1010**
 - 1111
14. Hva er inspeksjon i systemutvikling?
- Inspeksjon innebærer å finne feil eller forsikre seg om at feil ikke finnes uten å kjøre programmet**
 - Inspeksjon innebærer å undersøke om de som programmerer lager ryddige programmer som er lette å lese
 - Inspeksjon innebærer å finne feil i koden ved å kjøre programmet
15. Hva er en normal konfigurasjon av en klient-tjener arkitektur?
- Få tjenere, mange klienter**
 - Mange tjenere, få klienter
 - Mange tjenere, mange klienter
16. En bit av et program består av ei for-løkke inne i ei for-løkke. Begge løkkene løper igjennom tallverdiene 1:N. Tidsforbruket til denne programbiten vil være proporsjonalt med -
- N
 - $N \log N$
 - N^2
17. Gitt ei sortert liste av 272 verdier. Vi søker etter en verdi som ikke finnes i lista. Hvor mange sammenligninger må vi i verste fall gjøre ved bruk av binærsøk?
- 136
 - 8
 - 9**
18. Hva er en database i følge læreboka?
- En samling strukturerte data**
 - Et program for å håndtere store datamengder
 - Informasjon + metainformasjon
19. Hvilken av følgende alternativer blir **IKKE** i følge loven ansett som sensitive personopplysninger ved opprettelse av personregistre i følge læreboka?
- Religiøs bakgrunn
 - Karakterer**
 - Fagforeningsbakgrunn

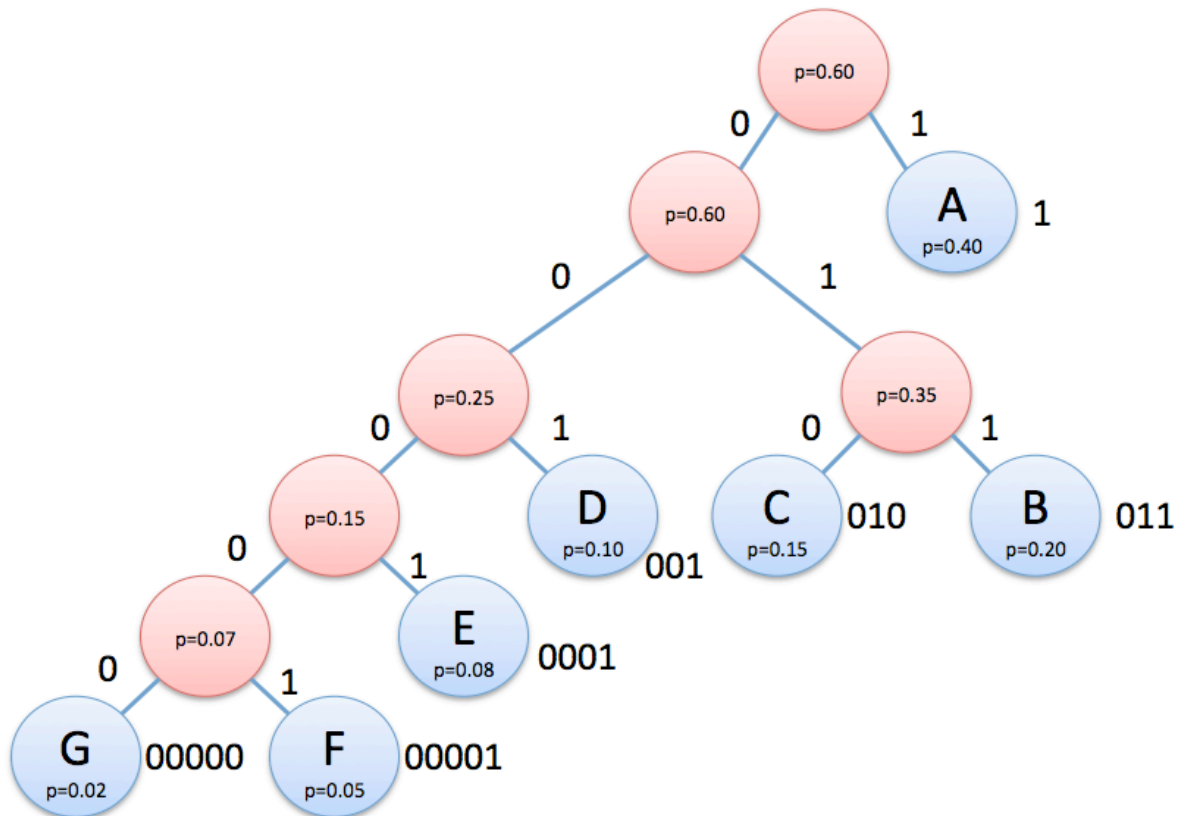
20. Bruk huffman-koding for å finne bitmønstrene til symbolene A, B, C, D, E, F og G når du har oppgitt fordeling som vist i Tabell 1 på en melding som består av 100 symboler.

Tabell 1 Fordeling av antall symboler oversendt i en melding på 100 symboler

Symbol	Antall i melding
A	40
B	20
C	15
D	10
E	8
F	5
G	2

- a) A:1, B:01, C:011, D:001, E:0011, F:0001, G:00011
- b) **A:1, B:011, C:010, D:001, E:0001, F:00001, G:00000**
- c) A:1; B:01, C:010, D:001, E:0010, F0001, G:0010

Løsning:



Oppgave 2 – Programmering (70 %)

I denne oppgaven skal du programmere metoder til et Snake spill, der spilleren skal styre en slange som beveger seg rundt på et spillbrett.

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	0	1	1	1	1	1	1
3	1	1	1	0	1	1	1	1	1	1
4	1	1	0	0	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Spillbrettet en kvadratisk tabell av en vilkårlig størrelse, der 1 indikerer en tom rute.

Selve ormen representeres av to tabeller for hver akse (x og y) som tilsvarer lengden på ormen.

På figuren til venstre er ormen tegnet inn med tegnet '0' og er av lengde 5.

Tabellen for denne ormen er:

```
orm_x = {3,3,3,2,1};
```

```
orm_y = {2,3,4,4,4};
```

Hodet på ormen er markert med mørkest farge og vil da være punktet (3,2). Halen vil være punktet (1,4).

Merk at halen er tegnet inn som en 1'er, slik at ormen visker ut etter seg hvis den beveger seg på brettet.

a) (5%) Skriv metoden *lag_tomt_brett* som genererer (lager) et tomt kvadratisk spillbrett (tabell bestående av bare 1'ere) av en gitt størrelse bestemt av en inn-parameter.

Løsning:

```
int[][] lag_tomt_brett(int storrelse) {
    int[][] tabell = new int[storrelse][storrelse];
    for (int i=0;i<storrelse;i++) {
        for (int j=0;j<storrelse;j++) {
            tabell[i][j] = 1;
        }
    }
    return tabell;
}
```

b) (10%) Skriv metoden *legg_ut_epler* som tar inn spillbrettet (en tabell av vilkårlig størrelse), samt antallet mat som skal legges ut vilkårlig på brettet. Mat blir representert av tallet 5.

I denne oppgaven skal metoden `Math.random` benyttes:

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);`

Løsning:

```
void legg_ut_epler(int[][] tabell, int antall) {
    while (antall>0) {
        int x = (int) (Math.random()*tabell.length);
        int y = (int) (Math.random()*tabell.length);
        if (tabell[y][x]==1) {
            tabell[y][x] = 5;
            antall = antall - 1;
        }
    }
}
```

c) (5%) Skriv metoden *sjekk_eple* som tar inn en x- og en y-posisjon (typisk hodet på ormen), samt spillbrettet. Metoden skal returnere *true* hvis et eple befinner seg på den angitte posisjonen (dvs. tallet 5), og *false* hvis det ikke gjør det.

Løsning:

```
boolean sjekk_eple(int x, int y, int[][] table) {
    if (table[y][x]==5) {
        return true;
    } else {
        return false;
    }
}
```

d) (15%) Skriv metoden *tegn_orm* som tar inn spillbrettet (en tabell av vilkårlig størrelse) samt de to tabellene som representerer ormen (*orm_x* og *orm_y*) og tegner ormen inn i spillbrettet representert med 0'ere. Halen på ormen skal tegnes som en 1'er.

Løsning:

```
void tegn_orm(int[][] table, int[] orm_x, int[] orm_y) {
    for (int i=0; i<orm_x.length-1;i++) {
        table[orm_y[i]][orm_x[i]] = 0;
        // Tegner 0 for ormen
    }
    table[orm_y[orm_y.length-1]][orm_x[orm_x.length-1]] = 1;
    // Tegner inn 1 for siste del av ormen
}
```

e) (15%) Skriv metoden *sjekk_krasj* som tar inn tabellene for ormen (*orm_x* og *orm_y*), samt retningen ormen skal bevege seg i (N, E, S eller V). Metoden skal sjekke om ormen krasjer i seg selv hvis den beveger seg i angitt retning. Metoden skal returnere *true* hvis den krasjer og *false* hvis den ikke gjør det.

Løsning:

```
boolean sjekk_krasj(int[] orm_x, int[] orm_y, String retning) {
    int sjekk_x=orm_x[0];
    int sjekk_y=orm_y[0];
    if (retning.equals("N")) {
        sjekk_y--;
    } else if (retning.equals("E")) {
        sjekk_x++;
    } else if (retning.equals("S")) {
        sjekk_y++;
    } else if (retning.equals("V")) {
        sjekk_x--;
    }
    boolean krasj = false;
    for (int i=0;i<orm_x.length-1;i++) {
        if (sjekk_x==orm_x[i] && sjekk_y==orm_y[i]) {
            krasj = true;
        }
    }
    return krasj;
}
```

f) (10%) Skriv metoden *flytt_or*m som tar inn tabellene for ormen (*orm_x* og *orm_y*) samt retningen ormen skal bevege seg (N, E, S eller V). Metoden skal flytte posisjonene i de to tabellene slik at ormen beveger seg i riktig retning. Hvis vi tar utgangspunkt i verdiene gitt i tabellene *orm_x* og *orm_y* ovenfor, så vil for eksempel en forflytning i retning N (Nord) gi tabellene: *orm_x* = {3,3,3,3,2} og *orm_y* = {1,2,3,4,4}

Løsning:

```
void flytt_orm(int[] orm_x, int[] orm_y, String retning) {
    int temp_x = orm_x[0]; // ta vare på fremste x-posisjon på ormen
    int temp_y = orm_y[0]; // ta vare på fremste y-posisjon på ormen
    for (int i=orm_x.length-1;i>0;i--) {
        orm_x[i] = orm_x[i-1]; // flytt ormen framover x-posisjon
        orm_y[i] = orm_y[i-1]; // flytt ormen framover y-posisjon
    }
    orm_x[0] = temp_x; // Gi tilbake orginal fremste posisjon x
    orm_y[0] = temp_y; // Gi tilbake orginal fremste posisjon y
    if (retning.equals("N")) {
        orm_y[0]--;
    } else if (retning.equals("E")) {
        orm_x[0]++;
    } else if (retning.equals("S")) {
        orm_y[0]++;
    } else if (retning.equals("V")) {
        orm_x[0]--;
    }
}
```

g) (10%) Bruk metodene angitt ovenfor for å skrive kode som gjør følgende (trenger ikke å definere en egen metode for denne koden):

1. Opprett en heltalls variabel *score* som tar vare på antall poeng i spillet og sett denne variabelen lik 0
2. Opprett et *spillbrett* (tabell) med størrelsen 10 x 10 og fyll tabellen med 0'er
3. Opprett tabellene *orm_x* og *orm_y* for å representere ormen med størrelse 3
4. Legg inn følgende punkter i ormen: Hodet er punktet (3,4), midten er punktet (2,4) og halen er punktet (1,4)
5. Legg ut 5 epler på spillbrettet
6. Sjekk om posisjonen til hodet på ormen på spillbrettet har mat (tallet 5), og øk variabelen *score* med 10 hvis dette er tilfellet
7. Sjekk om ormen krasjer i seg selv hvis den beveger seg nordover (N) og minsk variabelen *score* med 5 hvis dette er tilfelle
8. Flytt ormen en posisjon i retning nordover (N)
9. Tegn ormen inn i *spillbrett* (tabellen)
10. Skriv ut spillbrettet til skjerm ved hjelp av *out.println()*

Løsning:

```
<%
int score = 0; // #1
int hode = 0;

int[][] spillbrett = lag_tomt_brett(10); // #2
int[] orm_x = new int[3]; // #3
int[] orm_y = new int[3];
orm_x[0] = 3; orm_y[0] = 4; // #4
orm_x[1] = 2; orm_y[1] = 4;
orm_x[2] = 1; orm_y[2] = 4;

legg_ut_epler(spillbrett, 5); // #5

if(sjekk_eple(orm_x[hode],orm_y[hode], spillbrett)) { // #6
    score = score + 10;
}

if(sjekk_krasj(orm_x,orm_y,"N")) { // #7
    score = score - 5;
}

flytt_orm(orm_x,orm_y,"N"); // #8

tegn_orm(spillbrett, orm_x, orm_y); // #9

for (int i = 0 ; i<spillbrett.length; i++) { // #10
    for (int j = 0; j<spillbrett.length; j++) {
        out.println(spillbrett[i][j]);
    }
    out.println("<br>");
}
%>
```


Oppgave 4: HTML (10%)

I denne oppgaven får du oppgitt en html-fil med 10 mangler. Du skal skrive HTML-koden for å lage en webside som vist under. Merk følgende opplysninger:

1. Alle innslag i tabellen er i utgangspunktet (default) venstrejustert horisontalt og midtjustert vertikalt.
2. Bildet *geek.jpg* skal brukes i venstre kolonne og har størrelsen 310 x 310 pixels.
3. Bildet *header.jpg* skal brukes i midtkolonnen og har størrelsen 825 x 105 pixels.
4. Venstre kolonne skal ha bredde på 120 pixels
5. Header i midtkolonnen skal ha høyde på 70 pixels.

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

	Hjalmar Pjalle's kuule webside!!!	Dato: 25.mai 2010
Meny Verdens Gang Dagens Næringsliv Vårt Land Dagbladet NRK	<h2 style="text-align: center;">Velkommen til hjemmesiden min!</h2> <p>Her skal hoveddelen av websiden være. Her kommer det mye rart...</p> <p>Lurer på om studentene i ITGK klarer layouten... Håper det...</p>	
Ta kontakt på epost: hjalmar.pjalle@mysilgoerra.uff		

(oppgaven fortsetter på neste side)

Html-filen

1	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2	<html>
3	<head>
4	<title>Hjemmeside layout</title>
5	</head>
6	<body>
7	Fyll inn 1
8	<tr>
9	Fyll inn 2
10	Fyll inn 3
11	<h3>Meny</h3>
12	Verdens Gang
13	Dagens Næringsliv
14	Vårt Land
15	Dagbladet
16	NRK
17	</td>
18	<td>
19	Fyll inn 4
20	</td>
21	Fyll inn 5
22	Dato: 25.mai 2010
23	</td>
24	Fyll inn 6
25	<tr>
26	Fyll inn 7
27	<h1>Velkommen til hjemmesiden min!</h1>
28	<p>Her skal hoveddelen av websiden være. Her kommer det mye rart... </p>
29	<p>Lurer på om studentene i ITGK klarer layouten... Håper det...</p>
30	</td>
31	Fyll inn 8
32	<tr>
33	Fyll inn 9
34	Ta kontakt på epost: <u>hjalmar.pjalle@mysilgoerra.uff</u>
35	</td>
36	</tr>
37	Fyll inn 10
38	</body>
39	</html>

Løsning:

Html-filen

1	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2	<html>
3	<head>
4	<title>Hjemmeside layout</title>
5	</head>
6	<body>
7	<table border="1" cellpadding="5">
8	<tr>
9	<td rowspan="3" valign="top">
10	
11	<h3>Meny</h3>
12	Verdens Gang
13	Dagens Næringsliv
14	Vårt Land
15	Dagbladet
16	NRK
17	</td>
18	<td>
19	
20	</td>
21	<td align="right">
22	Dato: 25.mai 2010
23	</td>
24	</tr>
25	<tr>
26	<td colspan="2" valign="top">
27	<h1>Velkommen til hjemmesiden min!</h1>
28	<p>Her skal hoveddelen av websiden være. Her kommer det mye rart... </p>
29	<p>Lurer på om studentene i ITGK klarer layouten... Håper det...</p>
30	</td>
31	</tr>
32	<tr>
33	<td colspan="2" align="center">
34	Ta kontakt på epost: <u>hjalmar.pjalle@mysilgoerra.uff</u>
35	</td>
36	</tr>
37	</table>
38	</body>
39	</html>

Svarskjema flervalgsoppgave

Studentnummer: _____ Linje: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

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

Svarskjema flervalgsoppgave

Studentnummer: _____ Linje: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

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