

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultetet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk og
informasjonsvitenskap

BOKMÅL



Sensurfrist: xx. xx 2011

Kontinuasjoneksamen i TDT4105
Informasjonsteknologi, grunnkurs
XXXdag XX. august 2011
9.00 – 13.00

Faglig kontakt under eksamen:
Roger Midtstraum (995 72 420)

Hjelpemidler: C

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

Tilleggshefte II, "Introduksjon til: MATLAB"

Bestemt, enkel kalkulator: HP 30S eller Citizen SR270-X

Sensur:

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

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 (25 %)
- Oppgave 2: Programforståelse (10 %)
- Oppgave 3: Programmering (15 %)
- Oppgave 4: Programmering (50 %)

Oppgave 1: Flervalgsoppgave (25 %)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på det 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. Hva er en *tjener* i en klient-/tjenerarkitektur?

- a) En tjener utfører tjenester på vegne av en klient.
- b) En tjener etterspør tjenester fra klienter.
- c) En tjener er et program som vil ha noe utført på en annen maskin.
- d) Ingen av alternativene i a, b og c.

2. En del av et program består av en FOR-løkke, inne i en FOR-løkke, inne i en FOR-løkke. Alle løkkene løper gjennom tallene fra 1 til N . Tidsforbruket for denne programdelen vil være proporsjonal med:

- a) N^3
- b) $3N$
- c) $3N \log N$

3. Hva er hovedoppgaven til en *navnetjener*?

- a) Å oversette domenenavn til IP-adresser.
- b) Å oversette IP-adresser til domenenavn.
- c) Å holde rede på definerte variabelnavn når et program kjører.
- d) Ingen av alternativene i a, b og c.

4. Kari har utviklet et programsystem i den bedriften der hun har vært ansatt. Kan hun ta med seg og videreutvikle denne programvaren hvis hun slutter i bedriften?

- a) Ja, i følge Åndsverksloven er det programmereren som har rettighetene til programvaren.
- b) Nei, det er bedriften som eier programvaren.
- c) Det er uavklart siden dette er ikke regulert av lovgiving. Rettighetene til programvaren må avtales mellom partene i hvert enkelt tilfelle.

5. Svart-boks testing er:

- a) Testing der man prøver ut systemets eller modulens indre oppbygging.
- b) Testing der man ser bort fra systemets eller modulens indre oppbygging og kun tar i betraktning de ytre egenskapene.
- c) Testing der man ser bort fra systemenes eller modulenes indre oppbygging og prøver ut hvordan de ulike delene i et system fungerer sammen.

6. Hva er riktig rekkefølge når ulike minnetyper sorteres etter synkende hastighet?

- a) Register, cache, primærminne, sekundærminne.
- b) Cache, register, primærminne, sekundærminne.
- c) Primærminne, cache, sekundærminne, register.

7. Når kan man *ikke* benytte seg av binærsøkingsalgoritmen?

- a) Når datamengden er så stor at det vil ta for lang tid.
- b) Når det er stor sannsynlighet for at det man søker etter ikke finnes i datamengden.
- c) Når datamengden ikke er sortert.
- d) Man kan alltid bruke binærsøkingsalgoritmen.

8. Hva er i lovverket ikke definer som *sensitive* personopplysninger:

- a) Etnisk bakgrunn.
- b) Politisk oppfatning.
- c) Filosofisk oppfatning.
- d) Alle alternativene i a, b og c er sensitive personopplysninger.

9. Vi skal kode tegnene a-å, A-Å, 0-9, mellomrom, spørsmålstegn, utropstegn, punktum, kolon, semikolon og bindestrek. Hvor mange binære siffer (engelsk: bits) er nødvendig i kodene når alle tegn skal representeres med like mange binære siffer?

- a) 6
- b) 7
- c) 8

10. Hva blir resultatet når 81 kodes binært (i 2-tallsystemet)?

- a) 1011011
- b) 1010101
- c) 1010001
- d) Ingen av alternativene i a, b og c er riktige.

11. Hva tilsvarer ABCD i det heksadesimale tallsystemet (16-tallsystemet) i 10-tallsystemet?

- a) 42561
- b) 43981
- c) 44981
- d) Ingen av alternativene i a, b og c er riktige.

12. QoS er forkortelse for:

- a) Quantity of Storage.
- b) Quality of Servers.
- c) Quality of Service.
- d) Ingen av alternativene i a, b og c er riktige.

13. Vi har en sortert liste med 100 000 000 elementer. Ved binærsøking i denne listen, hvor mange sammenligninger må vi i verste fall gjøre?

- a) Omtrent 20.
- b) Omtrent 27.
- c) Omtrent 34.

14. `<p>...</p>` i en HTML-fil definerer:

- a) Et avsnitt.
- b) Et adressefelt.
- c) En hyperlenke.
- d) Ingen av alternativene i a, b og c.

15. $A \parallel \sim(B \ \&\& \ C)$ er usant (engelsk: false) når:

- a) A false, B true, C true
- b) A true, B true, C false
- c) A false, B false, C true
- d) Uttrykket er ikke usant for noen av alternativene i a, b og c.

16. Hva er *entropi*?

- a) Et mål på informasjoninnholdet i en melding.
- b) En algoritme for effektiv koding av symboler.
- c) En feiltilstand som gjør at lesehodet på en harddisk ikke klarer å følge sporene nøyaktig og derfor blander data fra flere spor.
- d) Ingen av alternativene i a, b og c er riktige.

17. En URL (Uniform Resource Locator) har følgende format:

- a) Protokoll, maskin, sti, fil
- b) Protokoll, maskin, sti
- c) Domene, sti, fil
- d) Ingen av alternativene i a, b og c er riktige.

18. Anta at karakterene har følgende fordeling: A (12 %), B (24 %), C (35 %), D (20 %), E (7 %) og F (2 %). Hva er en korrekt Huffmankoding for A-F?

- a) A: 100 B: 010 C: 001 D: 011 E: 101 F: 110
- b) A: 1100 B: 10 C: 0 D: 111 E: 11010 F: 11011
- c) A: 1111 B: 11 C: 1 D: 111 E: 11111 F: 111111
- d) Ingen av alternativene i a, b og c.

19. En IP-adresse er:

- a) En unik adresse for en datamaskin i Internettet.
- b) En mekanisme som gjør det mulig å kontakte en Internet Provider, for eksempel når det oppstår feil i nettet.
- c) Et felt som identifiserer data i en relasjonsdatabase.

20. Hva er *inspeksjon* av programvare?

- a) Kjøring av programmer for å se om programvaren oppfyller kravene til systemet.
- b) Overvåking av variablenes verdier under programkjøring for å finne årsakene til feil.
- c) Gjennomgang av programmer uten at de blir kjørt, med tanke på å finne feil og å forvisse seg om at de er riktige.

Oppgave 2: Programforståelse (10 %)

Gitt følgende programkode som er lagret i filen conv.m:

```
function res = conv(n,m)
i=1;
while (n > 0)
    temp(i) = rem(n,m);
    % rem returns remainder after division
    n = floor(n/m);
    % floor rounds towards minus infinity
    i = i + 1;
end % while

res = f(temp, i-1);

end % function

function res = f(list, n)

for i = 1:1:n
    res(i) = list(n-i+1);
end % for

end % function
```

Hva blir resultatet (i variablen result) når vi utfører koden:

```
result = conv(100,3);
```

Oppgave 3: Programmering (15 %)

Annuitetslån er lån der man betaler et like stort beløp i det antall terminer som lånet løper. Ved slike lån går mye av dette terminbeløpet til renter i starten, etter hvert blir en stadig større andel av terminbeløpet avdrag (tilbakebetaling).

I tabellen under har vi vist hvordan et annuitetslån på 1000 kroner nedbetales over 3 terminer med en terminrente på 5 %.

Termin	Restlån	Terminbeløp	Rente	Avdrag
0	1000.0			
1	682.8	367.2	50.0	317.2
2	349.7	367.2	34.1	333.1
3	0	367.2	17.5	349.7

Det som betales i rente i en termin regnes ut med formelen:

$$\text{restlån} * i$$

der restlån er gjestående lån ved inngangen av terminen og $i = \text{terminrente-i-prosent}/100$.

Det faste terminbeløpet (a) regnes ut med formelen:

$$a = \frac{P(1+i)(1 - (1/(1+i)^n))}{1 - (1/(1+i)^n)}$$

der P er lånebeløpet, i er terminrenten (terminrente-i-prosent/100) og n er antall terminer.

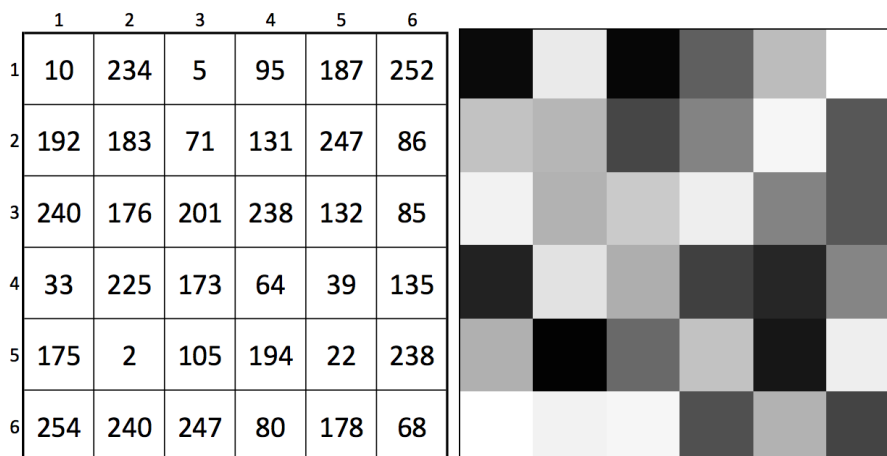
- (5 %) Skriv en funksjon *rente* som tar inn parametrene *laanBelop* og *renteFot* (terminrente i prosent), og returnerer renten som påløper i terminen.
- (5 %) Skriv en funksjon *termBelop* som tar inn parametrene *laanBelop*, *renteFot* og *antallTerminer*, og som returnerer beløpet som skal betales etter hver termin.
- (10 %) Skriv en funksjon *restLaan* som tar inn parametrene *laanBelop*, *renteFot*, *antallTerminer* og *termin*, og som returnerer restlånet etter at det har gått *termin* terminer. Dersom lånet er nedbetalt etter den aktuelle terminen, skal det returneres et restlån lik 0.

Dersom vi kaller opp `restLaan(1000, 5, 3, 2)` skal funksjonen returnere 349,7 som er restlånet etter to terminer i eksemplet som er vist i tabellen over.

Det er hensiktsmessig å gjenbruke funksjonene du laget i oppgave a og b. Du kan *bruke* funksjoner fra andre deloppgaver selv om du ikke har klart å løse deloppgaven der du skal lage funksjonen.

Oppgave 4: Programmering (50 %)

I denne oppgaven skal du programmere funksjoner brukt til analyse og behandling av kvadratiske bilder. Et bilde blir lagret i en kvadratisk tabell av tall, der hvert punkt i bilde kan ha verdi fra 0 til 255 (8-bits) som angir gråtonen i bildet. Verdien 0 angir fargen svart, verdien 255 angir fargen hvit, mens alle verdiene mellom angir gråtoner som blir lysere med høyere verdi. Figur 1 viser hvordan et bilde på størrelse 6x6 er representert som en tabell (til venstre i figuren) og hvordan et bilde med gråtoner generert fra tallene i tabellen ser ut (til høyre i figuren).



Figur 1: Gråtonebilde lagret som kvadratisk tabell (venstre) og vist som bilde (høyre)

I denne oppgaven kan følgende Matlab-funksjoner benyttes:

bar(x,y), bar(y)

produce bar graph of two vectors x and y
use indices of y as x-axis if one argument

floor(x)

return largest integer $\leq x$

imshow(A,[0,255]);

draws a picture of grey tones based on the contents of table A

rand, rand(N), rand(N,M)

return a random number on the open interval (0,1), a $N \times N$ or $N \times M$ matrix of random numbers

size(A), size(A,n)

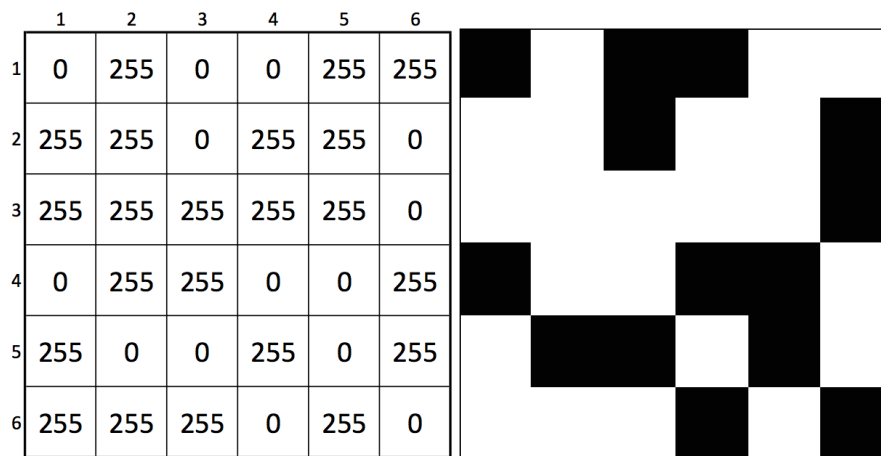
return all dimensions or nth dimension of A

zeros(N)

return $N \times N$ or $N \times M$ matrix of zeroes

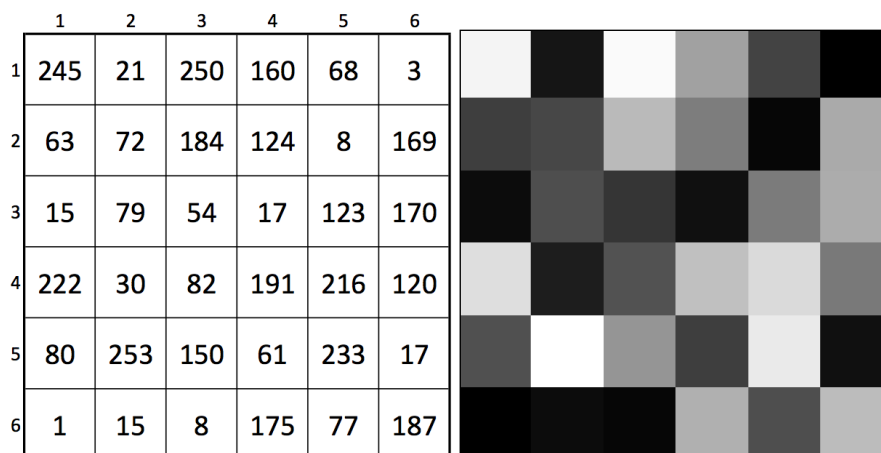
- (5%) Skriv funksjonen *random_picture* som tar inn parameteren *picsize*, og returnerer en kvadratisk tabell av tall på størrelse *picsize* som inneholder tilfeldige gråtoner (tall fra 0 til 255). Tabellen som er vist i figur 1 har *picsize* 6 etter som tabellen er en 6x6 matrise.
- (5%) Skriv funksjonen *test_picture* som tar inn en todimensjonal tabell av tall, *table*. Funksjonen skal returnere sannhetsverdien *sann* hvis tabellen er kvadratisk samt at tabellen ikke inneholder noen ulovlige tallverdier.
- (10%) Skriv funksjonen *filter_black_and_white* som tar inn en todimensjonal tabell av tall, *table*, samt et tall, *threshold*. Funksjonen skal returnere en endret tabell som er en svart/hvitt versjon av bildet representert i *table*. Alle verdiene i tabellen som er lavere enn

threshold skal få verdien 0 (svart), mens alle andre verdier skal få verdien 255 (hvit). Kalles funksjonen *filter_black_and_white* på tabellen fra figur 1 med *threshold* satt til 127, gir det resultatet som vist i figur 2.



Figur 2. Resultatet etter å ha kjørt *filter_black_and_white* på tabell fra figur 1

- d) (10%) Skriv funksjonen *filter_inverse*, som tar inn en todimensjonal tabell av tall, *table*, og returnerer en tabell der innholdet i tabellen er invertert, dvs. alt som er lyst på bilde skal bli mørkt og motsatt. Figur 3 viser resultatet etter å ha kjørt *filter_inverse* på tabellen fra figur 1.

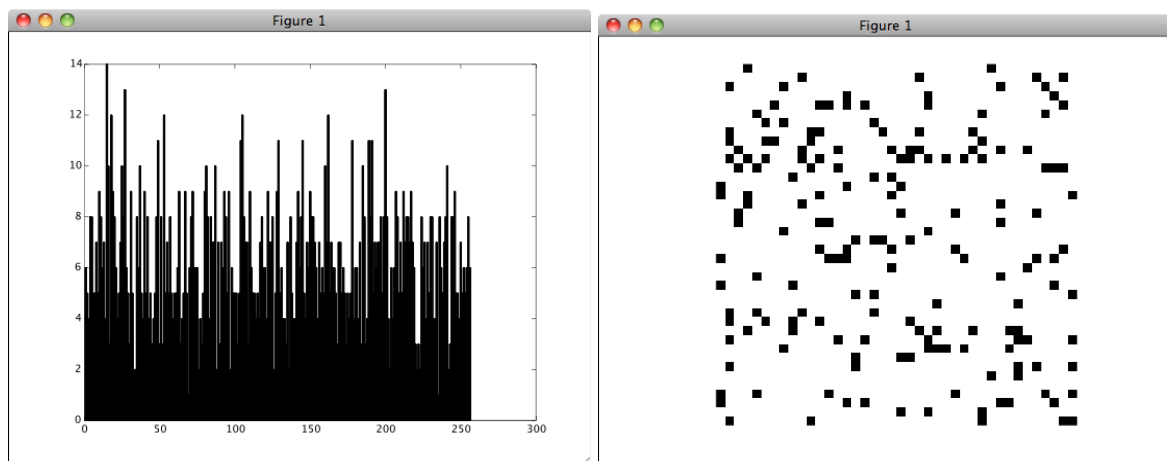


Figur 3. Resultat etter å ha kjørt *filter_inverse* på tabell fra figur 1.

- e) (10%) Skriv funksjonen *histogram* som tar inn en todimensjonal tabell av tall, *table*, og returnerer en endimensjonal tabell som inneholder antall punkter i bildet (*table*) som har en bestemt gråtone. For bildet i figur 3 vil histogrammet for eksempel vise at bildet har 1 punkt med verdi 1, 1 punkt med verdi 3, 2 punkter med verdi 8, 2 punkter med verdi 15, 2 punkter med verdi 17 osv.

- f) (10%) Bruk funksjonene ovenfor (selv om du ikke har løst deloppgavene) til å lage et Matlab-script som gjør følgende:
- Lag et bilde med tilfeldig innhold med størrelse 40x40 punkter.
 - Test om bildet er kvadratisk og at det ikke inneholder ulovlige verdier. Hvis bildet ikke blir godkjent, avslutt skriptet.
 - Inverter bildet.
 - Lag et histogram av bildet.
 - Vis histogram på skjerm som et søylediagram.
 - Spør brukeren om terskelverdi for sort/hvitt filtrering.
 - Gjør om bildet til sort/hvitt med terskelverdi gitt fra vi.
 - Tegn bilde på skjerm i gråtoner.

Eksempel på utskrift fra Matlab-scriptet, der bruker skrev inn terskelverdi 20:



Svarskjema flervalgsoppgave

Studentnummer: _____

Program: _____

Fagkode: _____

Dato: _____

Antall sider: _____

Side: _____

<i>Oppgave nr.</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</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: _____

Program: _____

Fagkode: _____

Dato: _____

Antall sider: _____

Side: _____

<i>Oppgave nr.</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</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				