

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultetet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk og
informasjonsvitenskap

BOKMÅL



Sensurfrist: 9. januar 2012

Avsluttende eksamen i TDT4105
Informasjonsteknologi, grunnkurs
Torsdag 8. desember 2011
9:00 – 13:00

Faglig kontakt under eksamen:

Roger Midtstraum, tlf 995 72 420
Svein-Olaf Hvasshovd, tlf 900 49 802
Asbjørn Thomassen, tlf 901 45 710

Hjelpemidler: C

Typegodkjent kalkulator: HP30S

Sensur:

Resultater gjøres kjent på studweb.ntnu.no og sensurtelefon 81548014.

Kvalitetssikret av:

Kai Torgeir Dragland, 28.11.2011

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: Grunnleggende programmering (25 %)
- Oppgave 3: Kodeforståelse (10 %)
- Oppgave 4: Mer programmering (40 %)
- Appendiks: Nyttige funksjoner
- Svarark til Flervalgsoppgave

Oppgave 1: Flervalgsoppgave (25 %)

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) Hva er et hovedkort (motherboard)?**
 - a) Et kretskort i en datamaskin der enheter som CPU, RAM, og andre enheter kobles sammen.
 - b) En minnekrets som tar vare på systemets innstillinger.
 - c) En prosessor for tynne mobiltelefoner.
 - d) Bunnplata i et PC kabinett.
- 2) Hva er en pakke (packet) i nettverkssammenheng?**
 - a) En datablokk av fast lengde som sendes gjennom nettverket, fra avsender til mottaker.
 - b) En datamelding som har ankommet og som står i kø for å bli levert til mottakermaskinen.
 - c) Den datamengden som utveksles mellom to datamaskiner som kommuniserer via nettverket.
 - d) Ingen av alternativene er riktig.
- 3) Hvilken av disse lagringsenhetene er ikke en sekundærlagrings-enhet?**
 - a) En harddisk.
 - b) En datamaskins hurtigbuffer (cache).
 - c) En minnepinne.
 - d) Alle alternativene er sekundærlagringsenheter.
- 4) Hva er en protokoll i nettverkssammenheng?**
 - a) Et sett kommunikasjonsregler for utveksling av data.
 - b) En avtale mellom nettverkseier og en bedrift som bruker nettet.
 - c) Et register der all nettverkstrafikk blir lagret i henhold til kravene fra myndighetene.
 - d) Ingen av alternativene er riktig.
- 5) Hvilket mål brukes vi vanligvis på overføringskapasitet i nettverk?**
 - a) Bits pr sekund (bps).
 - b) Gigabyte.
 - c) Båndbredde.
 - d) Ingen av alternativene er riktig.
- 6) Hva definerer et klient/tjener ("client/server") forhold?**
 - a) Klienter tilbyr data og tjenester til tjenere.
 - b) Klienter og tjenere tilbyr data og tjenester til hverandre.
 - c) Tjenere tilbyr data og tjenester til klienter.
 - d) Ingen av alternativene er riktig.
- 7) Hva definerer et "peer-to-peer" nettverk?**
 - a) En er sjef, de andre er slaver.
 - b) En er slave, de andre er sjefer.
 - c) Alle er likeverdige.
 - d) Ingen av alternativene er riktig.

8) Hvordan kan en GPS bestemme en posisjon?

- a) En GPS beregner sin posisjon ved å lokalisere nærmeste mobile basestasjon.
- b) En GPS beregner sin posisjon ved å bruke lokasjonen til flere satellitter.
- c) En GPS beregner sin posisjon ved å bruke lokasjon til kun en satellitt.
- d) Ingen av alternativene er riktig.

9) Hva er Wi-Fi?

- a) Et sett av standarder for trådløs dataoverføring.
- b) En kvalitetsbetegnelse for trådløse nett.
- c) Et mål på kvaliteten på en bredbåndabonnentslinje inn til huset.
- d) Ingen av alternativene er riktig.

10) En device driver er:

- a) en spesialdatamaskin for kjøretøy.
- b) spesialisert programvare for input/output, slik at utstyr kan kommunisere med resten av systemet.
- c) enheten som holder rede på neste instruksjon som skal utføres av en prosessor.
- d) Ingen av alternativene er riktig.

11) Et maskinspråk (machine language) er:

- a) et programmeringsspråk som oversettes av en kompilator (oversetter) til kjørbare kode.
- b) et binær-type programmeringsspråk bygd inn i prosessoren som datamaskinen kan kjøre direkte.
- c) er programmeringsspråk som er felles for alle datamaskiner slik at de kan kommunisere.
- d) Ingen av alternativene er riktig.

12) Ordstørrelse (word size) for en prosessor er:

- a) antall ord i en tekst som kan sammenlignes i et søk.
- b) antall bokstaver som kan behandles i en tekststreng.
- c) antall bit en prosessor kan prosessere på en gang.
- d) Ingen av alternativene er riktig.

13) Ytelse for superdatamaskiner måles i:

- a) FLOPS.
- b) Gigabytes.
- c) Antall prosessorkjerner.
- d) Ingen av alternativene er riktig.

14) Systemklokka i en datamaskin:

- a) fordeler tiden som brukes på ulike programmer.
- b) bestemmer hvor raskt operasjoner i en mikroprosessor utføres.
- c) sørger for at dato og tid alltid er riktig satt.
- d) Ingen av alternativene er riktig.

15) Hovedformålet med forstudiefasen (fase 1) i utvikling av informasjonssystemer er:

- a) Dokumentere krav til systemet.
- b) Programmere systemet.
- c) Gjennomføre en forberedende analyse.
- d) Ingen av alternativene er riktig.

16) Hva vil det si å vedlikeholde et informasjonssystem?

- a) Rette opp eksisterende feil i systemet.
- b) Utføre endringer i systemet basert på nye betingelser.
- c) Oppdatere dokumentasjon.
- d) Alle alternativene er riktig.

17) Hva gjør en enhetstest?

- a) Tester at ulike deler av systemet fungerer sammen på korrekt måte.
- b) Tester at selve datamaskinen (maskinvaren) fungerer.
- c) Tester individuelle deler av programvaren.
- d) Ingen av alternativene er riktig.

18) Hva er en algoritme?

- a) Krav som stilles til et dataprogram.
- b) En test for å finne feil i et dataprogram.
- c) En presis beskrivelse av operasjoner som skal utføres for å løse et problem.
- d) Ingen av alternativene er riktig.

19) Hva er et flytskjema?

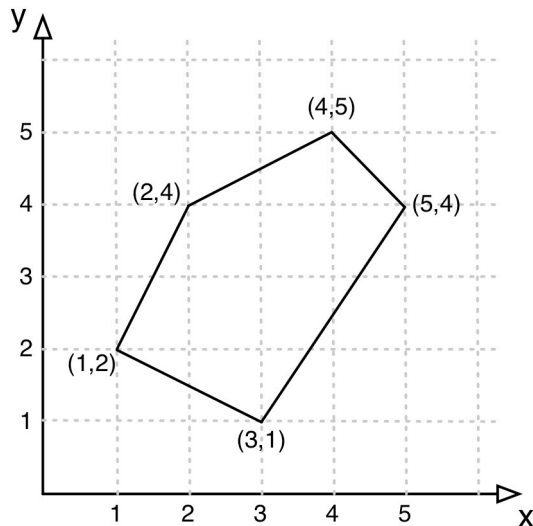
- a) Grafisk representasjon av en algoritme.
- b) Et skjema for å fylle inn informasjon på en webside.
- c) Et skjema som dokumenterer sikkerhet i et databasesystem.
- d) Ingen av alternativene er riktig.

20) Hva står ACID for innen databaser?

- a) Appropriate, Cynical, Isolation, Development.
- b) Appropriate, Collaborative, Irrelevant, Driver.
- c) Atomicity, Consistency, Isolation, Durability.
- d) Ingen av alternativene er riktig.

Oppgave 2 – Grunnleggende programmering (25%)

Figur 1 viser et eksempel på et polygon, en femkant. Vi kan representere et polygon som en liste (vektor) med alle hjørnekoordinatene, som vist i figur 2 for en femkant med hjørnepunktene (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) og (x_5, y_5) . Legg merke til at x-verdier og y-verdier alternerer gjennom listen og at antall elementer i listen vil variere med antall kanter i polygonet. Polygonet vist i figur 1 vil ha en punktliste som vist i figur 3.



Figur 1. Eksempel på et polygon

1	2	3	4	5	6	7	8	9	10
X_1	Y_1	X_2	Y_2	X_3	Y_3	X_4	Y_4	X_5	Y_5

Figur 2. Listerepresentasjon av et polygon

1	2	3	4	5	6	7	8	9	10
3	1	5	4	4	5	2	4	1	2

Figur 3. Listerepresentasjon av polygonet i figur 1.

Oppgave 2 a) (3 %)

Lengden på kanten mellom to hjørnepunkter, (x_i, y_i) og (x_{i+1}, y_{i+1}) , i et polygon er gitt av formelen:

$$\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

Skriv en funksjon `edgeLength` som tar inn koordinatene til to punkter som parametere og som returnerer lengden av kanten mellom punktene.

Dersom funksjonen kalles opp for punktene (3,1) og (5,4) i femkanten i figur 1, `edgeLength(3, 1, 5, 4)`, skal funksjonen returnere 3,61 ($\sqrt{13}$).

Oppgave 2 b) (8 %)

Omkretsen til et polygon er summen av kantlengdene i polygonet. Det finnes to spesialtilfeller. Et polygon med bare ett hjørnepunkt har omkrets lik 0, et polygon med to hjørnepunkter har omkrets lik 2 ganger kanten mellom punktene.

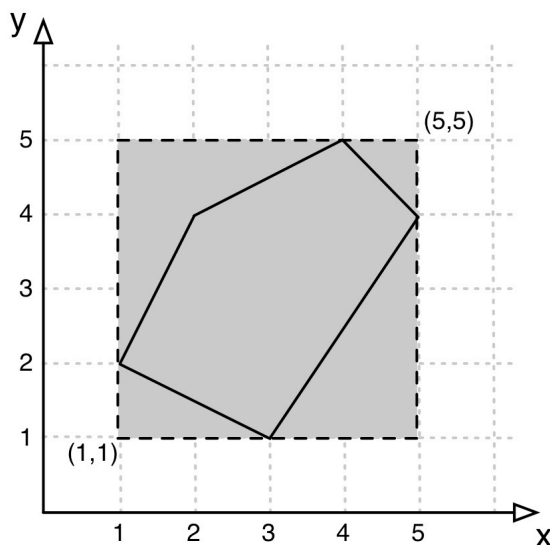
Skriv en funksjon *circumference* som tar inn *pList* som parameter og som returnerer omkretsen til polygonet som representeres av den aktuelle punktlisten. Dersom funksjonen kalles med en tom eller en ugyldig punktliste (et odde antall listeelementer) skal den returnere verdien -1.

Hvis funksjonen kalles opp med [3 1 5 4 4 5 2 4 1 2] som innparameter (femkanten i figur 4), skal den returnere verdien 11,7.

I denne oppgaven vil det være hensiktsmessig å gjenbruke funksjonen *edgeLength* fra deloppgave a. Du kan bruke denne funksjonen selv om du ikke fikk til å løse deloppgave a.

Oppgave 2 c) (6 %)

For et polygon kan vi beregne et omsluttende rektangel som akkurat inneholder polygonet. Figur 4 viser det omsluttende rektangelet til femkanten i figur 1. Legg merke til at kantene i det omsluttende rektangelet skal være parallelle med enten x- eller y-aksen.



Figur 4. Illustrasjon av omsluttende rektangel for et polygon

Det omsluttende rektangelet representeres med koordinatene til det nedre, venstre hjørnepunktet og koordinatene til det øvre, høyre hjørnepunktet.

Lag en funksjon *enclosingRectangle* som tar inn *pList* som parameter og som returnerer en vektor med koordinatene til det nedre, venstre hjørnepunktet og koordinatene til det øvre, høyre hjørnepunktet til det omsluttende rektangelet.

Dersom funksjonen kalles opp med [3 1 5 4 4 5 2 4 1 2] (femkanten i figur 4), skal den returnere vektoren [1 1 5 5].

Oppgave 2 d) (8 %)

Vi ønsker å kunne lese inn dataene for et polygon fra en tekstfil. Filen er ordnet slik at hver linje i filen inneholder x- og y-koordinatene for et hjørnepunkt i polygonet.

Skriv en funksjon *readPolygonFile* som tar inn et filnavn som parameter. Funksjonen skal lese hjørnepunktene fra filen og returnere en liste (vektor) med hjørnepunkter som beskrevet i starten av oppgaven.

Hvis vi har en fil med følgende innhold:

3	1
5	4
4	5
2	4
1	2

skal funksjonen returnere listen [3 1 5 4 4 5 2 4 1 2] (femkanten i figur 4). Du kan forutsette at alle linjer i tekstfilen inneholder to tall. Dersom det oppstår feil under filbehandlingen, skal funksjonen returnere -1.

Oppgave 3 – Kodeforståelse (10%)

Oppgave 3 a) (4 %)

Hva returneres hvis funksjonen `mystery5([3 4 6 7],[3 4 7 6],4)` med kode som vist under kjøres?

```
function res = mystery5(a,b,n)
%
res = 0;
x = 1;
while n>0
    if a(x)==b(x)
        res = res + 1;
    end
    x = x + 1;
    n = n - 1;
end
end
```

Oppgave 3 b) (3 %)

Forklar med *kun en kort setning* hva funksjonen `mystery5` gjør.

Oppgave 3 c) (3 %)

Hva returneres hvis funksjonen `mystery6(3,2)` med kode som vist under kjøres?

```
function res = mystery6(n,x)
%
res = 0;
if n>0
    res = x * mystery6(n-1,x);
else
    res = 1;
end
end
```


Oppgave 4 – Programmering (40 %)

I denne oppgaven skal du programmere ulike funksjoner som skal brukes til å tilby highscore-funksjonalitet i et dataspill. Highscore-lista skal kunne ta vare på de 10 beste highscorene bestående av deres poengsum og telefonnummer. Highscore-lista skal representeres av en todimensjonal tabell som vist i figuren under:

Indeks	Telefonnr	Poengsum
1	22048700	100
2	23313050	90
3	73595000	80
4	22000000	70
5	23048000	60
6	81544000	50
7	73594485	40
8	73590770	30
9	73593676	20
10	73591839	10

Bruk funksjoner som defineres i andre deloppgaver hvis mulig. Du kan bruke funksjoner fra andre deloppgaver selv om deloppgaven ikke er løst.

Oppgave 4 a) (5 %)

Skriv funksjon `check_highscore` som tar inn en poengsum (*points*) og en highscore-liste (*scores*) og returnerer plassen poengsummen får på highscore-lista (fra plass 1 til 10). Merk at poengsummen må være høyere enn et innslag på lista for å kapre plassen. Hvis poengsummen ikke er høyere enn noen av innslagene i lista, skal verdien -1 returneres.

Oppgave 4 b) (5 %)

Skriv funksjonen `print_highscores` som tar inn en highscore-liste (*scores*) og skriver ut til skjerm alle highscores med plassering, telefonnummer og poengsum som vist under. Merk at *plassering* skal skrives ut høyrejustert med 2 tegns feltbredde, og *poeng* høyrejustert med 5 tegns feltbredde.

```

1. 22048700    100
2. 23313050    90
3. 73595000    80
4. 22000000    70
5. 23048000    60
6. 81544000    50
7. 73594485    40
8. 73590770    30
9. 73593676    20
10. 73591839   10

```

Oppgave 4 c) (10 %)

Skriv funksjonen `add_highscore` som tar inn en poengsum (*points*) og et telefonnummer (*number*) og en highscore-liste (*scores*), og legger til poengsum og telefonnummer i highscore-lista hvis poengsummen er høy nok. Merk at det er kun innslaget med laveste poengverdi som skal ut av lista når en ny score blir lagt til. Funksjonen skal returnere highscore-lista som kan være enten uendret eller endret.

Figuren under viser highscore-lista før og etter at

`add_highscore(65,90909090,highscores)` er kjørt:

Før:		Etter:
1	22048700 100	1
2	23313050 90	2
3	73595000 80	3
4	22000000 70	4
5	23048000 60	5
6	81544000 50	6
7	73594485 40	7
8	73590770 30	8
9	73593676 20	9
10	73591839 10	10

`add_highscore(65,90909090,highscore)`

Oppgave 4 d) (10 %)

Skriv funksjonen `most_highscores` som tar inn en highscore-liste (*scores*) og returnerer nummeret til den person som har flest innslag på lista. Hvis det er flere med like mange innslag, skal funksjonen returnere telefonnummeret til den av spillerne som er lengst oppe på lista. Hvis lista inneholder kun 10 forskjellige nummer, skal tallet 0 returneres.

Oppgave 4 e) (10 %)

Skriv funksjonen `new_highscorelist` som returnerer en ny highscore-liste (todimensjonal tabell) med poengsummer fra 100 ned til 10 (100, 90, 80...) der følgende ti telefonnummer skal plasseres tilfeldig i highscore-lista (merk at alle telefonnummer skal representeres i lista): 1100000, 44000000, 22000000, 90909090, 73500000, 73000000, 22220000, 54000000, 30303030, 40404040.

Appendiks: Nyttige funksjoner

FIX Round towards zero.

FIX(X) rounds the elements of X to the nearest integers towards zero.

FLOOR Round towards minus infinity.

FLOOR(X) rounds the elements of X to the nearest integers towards minus infinity.

FCLOSE Close file.

ST = FCLOSE(FID) closes the file associated with file identifier FID, which is an integer value obtained from an earlier call to FOPEN. FCLOSE returns 0 if successful or -1 if not.

FEOF Test for end-of-file.

ST = FEOF(FID) returns 1 if the end-of-file indicator for the file with file identifier FID has been set, and 0 otherwise. The end-of-file indicator is set when a read operation on the file associated with the FID attempts to read past the end of the file.

FGETL Read line from file, discard newline character.

TLINE = FGETL(FID) returns the next line of a file associated with file identifier FID as a MATLAB string. The line terminator is NOT included. Use FGETS to get the next line with the line terminator INCLUDED. If just an end-of-file is encountered, -1 is returned.

FOPEN Open file.

FID = FOPEN(FILENAME,PERMISSION) opens the file FILENAME in the mode specified by PERMISSION:

- 'r' open file for reading
- 'w' open file for writing; discard existing contents
- 'a' open or create file for writing; append data to end of file
- 'r+' open (do not create) file for reading and writing
- 'w+' open or create file for reading and writing; discard existing contents
- 'a+' open or create file for reading and writing; append data to end of file

FPRINTF Write formatted data to file.

COUNT = FPRINTF(FID,FORMAT,A,...) formats the data in the real part of array A (and in any additional array arguments), under control of the specified FORMAT string, and writes it to the file associated with file identifier FID. COUNT is the number of bytes successfully written. FID is an integer file identifier obtained from FOPEN. It can also be 1 for standard output (the screen) or 2 for standard error. If FID is omitted, output goes to the screen.

FORMAT is a string containing ordinary characters and/or C language conversion specifications. Conversion specifications involve the character %, optional flags, optional width and precision fields, optional subtype specifier, and conversion characters d, i, o, u, x, X, f, e, E, g, G, c, and s.

The special formats \n,\r,\t,\b,\f can be used to produce linefeed, carriage return, tab, backspace, and formfeed characters respectively. Use \\ to produce a backslash character and %% to produce the percent character.

LENGTH Length of vector.

LENGTH(X) returns the length of vector X. It is equivalent to MAX(SIZE(X)) for non-empty arrays and 0 for empty ones.

MOD Modulus after division.

MOD(x,y) is $x - n \cdot y$ where $n = \text{floor}(x./y)$ if $y \neq 0$.

RAND Uniformly distributed pseudorandom numbers.

R = RAND(N) returns an N-by-N matrix containing pseudorandom values drawn from the standard uniform distribution on the open interval(0,1). RAND(M,N) or RAND([M,N]) returns an M-by-N matrix.

RANDI Pseudorandom integers from a uniform discrete distribution.

R = RANDI(IMAX,N) returns an N-by-N matrix containing pseudorandom integer values drawn from the discrete uniform distribution on 1:IMAX. RANDI(IMAX,M,N) or RANDI(IMAX,[M,N]) returns an M-by-N matrix.

REM Remainder after division.

REM(x,y) is $x - n \cdot y$ where $n = \text{fix}(x./y)$ if $y \neq 0$.

SIZE Size of array.

D = SIZE(X), for M-by-N matrix X, returns the two-element row vector D = [M,N] containing the number of rows and columns in the matrix.

SQRT Square root.

SQRT(X) is the square root of the elements of X.

STR2NUM Convert string matrix to numeric array.

X = STR2NUM(S) converts a character array representation of a matrix of numbers to a numeric matrix. For example,
S = ['1 2' ; '3 4'] str2num(S) => [1 2;3 4]

SUM Sum of elements.

S = SUM(X) is the sum of the elements of the vector X. If X is a matrix, S is a row vector with the sum over each column.

Svarskjema flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

<i>Oppgavenr</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

Kandidatnummer: _____

Program: _____

Fagkode: _____

Dato: _____

Antall sider: _____

Side: _____

<i>Oppgavenr</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				