

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4105 Informasjonsteknologi, grunnkurs kontinuasjonseksamen

Faglig kontakt under eksamen:

Tlf.:

Eksamensdato:

Eksamenstid (fra-til): 09:00 – 13:00

Hjelpemiddelkode/Tillatte hjelpemidler: Godkjent kalkulator

Annen informasjon:

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.

Målform/språk: Bokmål

Antall sider:

Innhold:

- Oppgave 1: Flervalgsoppgave (30 %)
- Oppgave 2: Grunnleggende programmering (10 %)
- Oppgave 3: Kodeforståelse (10 %)
- Oppgave 4: Mer programmering (50 %)
- Appendiks: Nyttige funksjoner
- Svarark til Flervalgsoppgave

Kontrollert av:

Dato

Sign

Oppgave 1: Flervalgsoppgave (30 %)

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 pseudokode?**
 - a. Foreløpig kode som ikke skal brukes i den endelige versjonen av et system
 - b. En diagramteknikk for å beskrive algoritmer eller programmer
 - c. De delene av et program som kan være årsak til en feil som er avdekket under testing
 - d. En måte å beskrive algoritmer som er mindre formell enn regulære programmeringsspråk
- 2) **Vi har 43 ulike symboler som vi ønsker å kode digitalt. Hva er det minste antall bits som vi må bruke?**
 - a. 5
 - b. 6
 - c. 7
 - d. 8
- 3) **Hvilken av disse aksessteknologiene gir høyest kapasitet ?**
 - a. Edge
 - b. 3G/UMTS
 - c. HSDPA
 - d. LTE
- 4) **Retningslinjene for algoritmer sier at det lønner seg å skrive en løkke for å utføre en repeterende oppgave hvis oppgaven må utføres**
 - a. mer enn 1 gang
 - b. 3-5 ganger eller mer
 - c. 8-10 ganger eller mer
 - d. 20 ganger eller mer
- 5) **Vi skal bruke binærsøking i en datamengde med 1 000 000 sorterte elementer. Hvor mange oppslag må vi vanligvis gjøre?**
 - a. 10
 - b. 20
 - c. 30
 - d. 50
- 6) **Analog informasjon er**
 - a. Diskret
 - b. Kontinuerlig
 - c. Tilfeldig
 - d. Digital
- 7) **Det er fire grunnleggende aktiviteter som inngår i alle programvareutviklingsprosesser.**
 - a. Spesifikasjon, utvikling, validering, evolusjon
 - b. Spesifikasjon, analyse, programmering, bruk
 - c. Forstudie, implementasjon, testing, bruk
 - d. Forstudie, programmering, simulering, vedlikehold

- 8) **Hvor mange negative og positive heltall kan vi representere i 2 byte?**
- 2048
 - 32768
 - 65536
 - 1048576
- 9) **Hvorfor kan et WiFi nettverk påvirke en Bluetooth forbindelse ?**
- WiFi nettverk har så sterkt signal
 - Bluetooth og WiFi bruker samme frekvensbånd
 - Bluetooth og WiFi bruker samme metode for koding
 - Ved samtidig bruk av WiFi og Bluetooth dannes et fenomen som heter "stående bølger"
- 10) **Hvor mange instruksjoner er vanlig i en moderne datamaskin?**
- 3-7
 - 10-20
 - rundt 100
 - tusenvis
- 11) **Hva kjennetegner problemløsning etter top-down prinsippet?**
- Vi løser de vanskeligste problemene først
 - Vi designer før vi programmerer en løsning
 - Vi går fra det generelle til det spesielle
 - Vi løser de enkelte delene av problemet først og bruker disse løsningene som byggeklosser i en generell løsning
- 12) **Sortering ved innsetting (insertion sort) bruker omtrent 1 sekund på å sortere 1000 elementer. Hvor lang tid vil det omtrent ta å sortere 10000 elementer?**
- 2 s
 - 10 s
 - 100 s
 - 1000 s
- 13) **Hva er pipelining?**
- Lignende operasjoner blir utført i rekkefølge
 - Flere instruksjoner kan være under utførelse samtidig
 - En effektiv organisering av datamaskinens hukommelse
 - En mekanisme for å utveksle data mellom programmer
- 14) **Hvor mange symboler kan representeres av en sekvens på K bit?**
- K^2
 - 2^K
 - K
 - 2
- 15) **Et nettverk sin evne til å levere uavbrutt / kontinuerlig tjeneste defineres som**
- Skalerbarhet
 - Tilgjengelighet
 - Pålitelighet
 - Ytelse
- 16) **Ved digitalisering av lyd, hva oppnår vi med å bruke flere bit i hver punktprøve (sample)?**
- Vi trenger ikke å punktprøve (sample) like ofte
 - Vi oppfyller kravene i Nyquist-regelen
 - Vi kan representere stereo-lyd
 - Vi får en mer nøyaktig digitalisering

17) Det å utgi seg for være noen andre på internett kalles

- a. Phishing
- b. Spoofing
- c. Pharming
- d. Aliasing

18) Er det horisontale eller vertikale fargebånd som komprimeres bra i GIF-filer

- a. Vertikale
- b. Horisontale
- c. GIF-filer komprimerer hverken vertikale eller horisontale fargebånd spesielt bra
- d. GIF-filer komprimerer både horisontale og vertikale fargebånd veldig bra

19) Hente/Utføre-kretsløpet (Fetch/Execute Cycle) i en moderne datamaskin utføres

- a. En gang i sekundet
- b. Tusenvis av ganger i sekunder
- c. Hundretusenvis av ganger i sekundet
- d. Ikke oftere enn antall ganger klokken i maskina tikker

20) Hva kjennetegner inkrementell programvareutvikling?

- a. Utvikler programvaren de klart avskilte fasene krav, design, implementasjon, testing osv.
- b. Utvikler programvaren uten noen plan eller prosess
- c. Deler opp systemet i flere mindre deler som blir utviklet del for del
- d. Lager en rask prototype basert på eksisterende krav

Oppgave 2: Grunnleggende programmering (10 %)

Oppgave 2a (5%)

Lag funksjonen `yatzy`. Den skal ha 5 innparametere, kalt `t1`, `t2`, `t3`, `t4` og `t5`. Innparametrene representerer 5 tall mellom 1 og 6 (5 terninger).

Funksjonen skal returnere en liste som inneholder de 5 tallene i sortert rekkefølge, eller en feilmelding hvis en av tallene er større enn 6 eller mindre enn 1.

Eksempel på kjøringer:

```
>> yatzy(1,4,6,6,1)
ans =
     1     1     4     6     6

>> yatzy(1,4,6,4,10)
ans =
Ikke input verdier større enn 6!

>> yatzy(1,4,6,0,1)
ans =
Ikke input verdier mindre enn 1!
```

Oppgave 2b (5%)

Lag funksjonen `maxi_yatzy`. Den skal ta inn en liste med 5 eller 6 tall, og den skal returnere en skriftlig melding til brukeren som sier hvor mange terninger som ble kastet, hvilken verdi det var flest av, og hvor mange like det var av den verdien. Hvis det blir «uavgjort mellom to tall» brukes det høyeste tallet, slik som i dette eksempelet:

```
>> maxi_yatzy([1,2,3,4,4,3])
ans =
Du kastet 6 terninger, og fikk flest 4-ere, med 2 like.
```

Oppgave 3: Kodeforståelse (10 %)

Oppgave 3a (5%)

Gitt funksjonen `secret` vist under:

```
function y = secret(x)

    n = length(x);

    for i = 1:1:n
        a = x(1);
        k = 1;
        for j = 2:1:n
            if a < x(j)
                a = x(j);
                k = j;
            end
        end

        y(i) = a;
        x(k) = -1;
    end

end
```

Hva returneres av etter å ha utført:

```
secret([1 4 8 2 5 8 10 1])
```

Oppgave 3b (5%)

Gitt funksjonen `secret2` vist under:

```
function y = secret2(x)

    a = 0;

    while x > 0
        b = rem(x,10);
        a = a + 1;
        x = (x - b) / 10;
    end

    y = a;

end
```

Hva returneres av etter å ha utført:

```
secret2(12345678)
```

Oppgave 4: Mer programmering (50 %)

Oppgave 4a (5%)

Lag en funksjon, *enterWords*, som har ingen innparametre, og returnerer ei liste, *wordList*, med ord som brukeren selv har skrevet inn. Brukeren kan avslutte innskriving av ord ved å trykke enter uten å skrive noe. Prompten til brukeren skal vise "Enter word [press Enter to quit]:"

Eksempel på kjøring (ord med med fet skrift er ord brukeren selv har skrevet inn):

```
>> list = enterWords()
Enter word [Press Enter to quit]: house
Enter word [Press Enter to quit]: chair
Enter word [Press Enter to quit]: university
Enter word [Press Enter to quit]: mouse
Enter word [Press Enter to quit]:

list =
    'house'    'chair'    'university'    'mouse'
```

Oppgave 4b (10%)

Lag en funksjon, *noVowels*, som tar inn ei liste med ord (*inList*), og returnerer en ny liste, *outList*, med ord der alle vokalene i det engelske alfabetet (bokstavene "a", "e", "i", "o", "u" og "y") fra *inList* er fjernet.

Eksempel på kjøring (list har samme verdi som i oppgave 4a):

```
newlist = noVowels(list)

newlist =
    'hs'    'chr'    'nvrst'    'ms'
```

Oppgave 4c (10%)

Lag en funksjon, *randomSequence*, som tar inn to like lange lister (*listOne*, *listTwo*) og returnerer to lister (*newlistOne*, *newlistTwo*) der rekkefølgen på elementene i begge listene tilfeldig endret på samme måte. Det vil si at f.eks. element 3 i både *listOne* og *listTwo* for blir plassert som element 1 i både *newlistOne* og *newlistTwo*.

Eksempel på kjøring (*list* og *newlist* har samme verdier som i oppgave 4a/4b):

```
>> [answers, puzzles] = randomSequence(list, newlist)

answers =
    'chair'    'mouse'    'house'    'university'

puzzles =
    'chr'    'ms'    'hs'    'nvrst'
```

Oppgave 4d (5%)

Lag en funksjon `printNewLines` som tar inn en variabel, `number`, og skriver ut `number` antall linjeskift til konsoll.

Eksempel på kjøring:

```
>> printNewLines(5)
```

```
>>
```

Oppgave 4e (10%)

Lag funksjonen `playGame` som tar inn to lister (`answers`, `puzzles`), der brukeren skal bli presentert for ett og ett ord i lista `puzzles` (ord uten vokaler), gjette hvilket ord det er som sjekkes mot ord i lista `answers`. Riktige svar belønnes med 1 poeng. Funksjonen skal returnere antall poeng brukeren scorer i variabelen `points`. Hva som skal skrives ut til skjerm er vist i eksempelet på kjøring under. Ord i eksemplet skrevet med fet skrift er ord skrevet inn av brukeren.

Eksempel på kjøring (verdier fra listene er hentet for tidligere oppgaver):

```
>> points = playGame(answers, puzzles)
Puzzel word: chr
Guess word? chair
You answered correctly!
Puzzel word: ms
Guess word? mouse
You answered correctly!
Puzzel word: hs
Guess word? hose
You answered incorrectly! The answer should be house
Puzzel word: nvrst
Guess word? university
You answered correctly!

points =
```

```
3
```


Oppgave 4f) (10%)

Skrive koden for å gjøre ferdig spillet *NoVowels* game. Spillet skal gjøre følgende:

1) Skrive ut følgende til skjerm:

```
The NoVowels Game
=====
```

```
Player 2: Look away from the screen
```

```
Player 1: Write in a list of English words in lower-case.
```

2) Hente ei liste med ord som brukeren selv skriver inn og lagre dette i variabelen *wordList*.

3) Lage en ny liste, *noVowelsList*, som tilsvare lista *wordList* uten vokaler.

4) Lage to nye lister, *answers* og *quizzes*, der rekkefølgen på ordene i listene *wordList* og *noVowelsList* er byttet om vilkårlig, men på samme vis for begge listene.

5) Skriv ut 50 linjeskift til konsoll

6) Skriv ut til skjerm følgende:

```
Player 2: Guess words that lack all vowels:
```

7) Spill igjennom gjetting av ord med listene *answers* og *quizzes* og lagre poengsummen i variabelen *points*.

8) Skriv ut til skjerm hvor mange poeng spilleren fikk av antall mulige:

```
You have got 3 of 4 points
```

Utskrift av komplett kjøring (tekst fra bruker med fet skrift):

```
>> NoVowelsGame
```

```
The NoVowels Game
```

```
=====
```

```
Player 2: Look away from the screen
```

```
Player 1: Write in a list of English words in lower-case.
```

```
Enter word [Press Enter to quit]: house
```

```
Enter word [Press Enter to quit]: chair
```

```
Enter word [Press Enter to quit]: university
```

```
Enter word [Press Enter to quit]: mouse
```

```
Enter word [Press Enter to quit]:
```

```
(50 new lines here)
```

```
Player 2: Guess words that lack all vowels:
```

```
Puzzel word: chr
```

```
Guess word? chair
```

```
You answered correctly!
```

```
Puzzel word: ms
```

```
Guess word? mouse
```

```
You answered correctly!
```

```
Puzzel word: nvrst
```

```
Guess word? university
```

```
You answered correctly!
```

```
Puzzel word: hs
```

```
Guess word? host
```

```
You answered incorrectly! The answer should be house
```

```
You have got 3 of 4 points
```

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				

Løsningsforslag

Oppgave 1 (30%)

<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				

Oppgave 2 (10%)

Oppgave 2a (5%)

```
function liste = yatzy(t1, t2, t3, t4, t5)
    liste = [t1 t2 t3 t4 t5];
    liste = sort(liste);
    if max ( liste ) >6
        liste = 'Ikke bruk input større enn 6!';
    elseif min ( liste ) <1
        liste = 'Ikke bruk input mindre enn 1!';
    end
end %function
```

Oppgave 2b (5%)

```
function melding = maxi_yatzy(liste)
    flestverdi=0;
    maxantall=0;

    for i=1:6
        antall = sum(liste==i);
        if antall >= maxantall
            flestverdi = i;
            maxantall = antall;
        end
    end
    melding = sprintf('Du kastet %i terninger, og fikk flest %i-ere (%i
like).\n',...
        length(liste), flestverdi, maxantall);
end %function
```

Oppgave 3 (10%)

Oppgave 3a (5%)

```
[10 8 8 5 4 2 1 1]
```

Oppgave 3b (5%)

```
8
```

Oppgave 4 (50%)

Oppgave 4a (5%)

```
function list = enterWords()
    index = 1;
    result = 'something'; % To start the while loop
    while length(result) > 0
        result = input('Enter word [Press Enter to quit]: ', 's');
        if length(result) > 0
            list{index} = result;
            index = index + 1;
        end % if
    end % while
end % function
```

Oppgave 4b (10%)

```
function newList = noVowels(list)
    for i = 1:length(list)
        word = list{i};
        newWord = '';
        index = 1;
        for j = 1:length(word)
            if ~((word(j) == 'a') || (word(j) == 'e') || (word(j) == 'i') || ...
                (word(j) == 'o') || (word(j) == 'u') || (word(j) == 'y'))
                newWord(index) = word(j);
                index = index + 1;
            end % for
        end % for
        newList{i} = newWord;
    end % for
end % function
```

Oppgave 4c (10%)

```
function [newListOne, newListTwo] = randomSequence(listOne, listTwo)
    toIndex = 1;
    while length(listOne) > 0
        fromIndex = randi(length(listOne));
        newListOne{toIndex} = listOne{fromIndex};
        newListTwo{toIndex} = listTwo{fromIndex};
        listOne(fromIndex) = [];
        listTwo(fromIndex) = [];
        toIndex = toIndex + 1;
    end % while
end % function
```


Oppgave 4d (5%)

```
function printNewLines(number)
    for i = 1:1:number
        fprintf('\n');
    end % for
end % function
```

Oppgave 4e (10%)

```
function points = playGame(answers, puzzels)
    points = 0;
    for i = 1:1:length(answers)
        fprintf('Puzzel word: %s\n', puzzels{i})
        answer = input('Guess word? ', 's');
        if (strcmp(answer, answers{i}))
            fprintf('You answered correctly!\n')
            points = points + 1;
        else
            fprintf('You answered incorrectly! The answer should be %s\n',
answers{i})
        end % if
    end % for
end %
```

Oppgave 4f (10%)

```
fprintf('The NoVowels Game\n')
fprintf('=====\n')

fprintf('Player 2: Look away from the screen\n')
fprintf('Player 1: Write in a list of English words in lower-case.\n')

% Get a list of words entered by player 1
wordList = enterWords();

% Create a new list without vowels
noVowelsList = noVowels(wordList);

% Randomize the sequence of words for both lists
[answers, quizzes] = randomSequence(wordList, noVowelsList);

% Print 50 newlines
printNewLines(50);

% Play the game
fprintf('Player 2: Guess words that lack all vowels:\n')
points = playGame(answers, quizzes);
fprintf('You have got %d of %d points\n', points, length(answers))
```