NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultetet for informasjonsteknologi, matematikk og elektroteknikk

Institutt for datateknikk og informasjonsvitenskap

BOKMÅL

Sensurfrist: 9.januar 2012

Løsningsskisse til avsluttende eksamen i TDT4110 Informasjonsteknologi, grunnkurs

Torsdag 8. desember 2011
9:00 – 13:00

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 kjørbar 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?

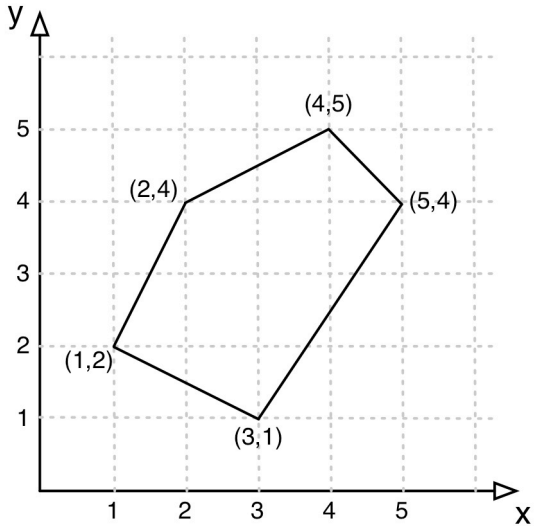
a) Grafisk representasjon av en algoritme.
b) Et skjema for å fylle inn informasjon på en webside.
c) Et skjema som dokumenterer sikkerhet i et databasesystem.
d) Ingen av alternativene er riktig.

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_0,y_0), (x_1,y_1), (x_2,y_2), (x_3,y_3) og (x_4,y_4). 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

0	1	2	3	4	5	6	7	8	9
X_0	Y_0	X_1	Y_1	X_2	Y_2	X_3	Y_3	X_4	Y_4

Figur 2. Listerepresentasjon av et polygon

0	1	2	3	4	5	6	7	8	9
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 ($\sqrt{13}$).

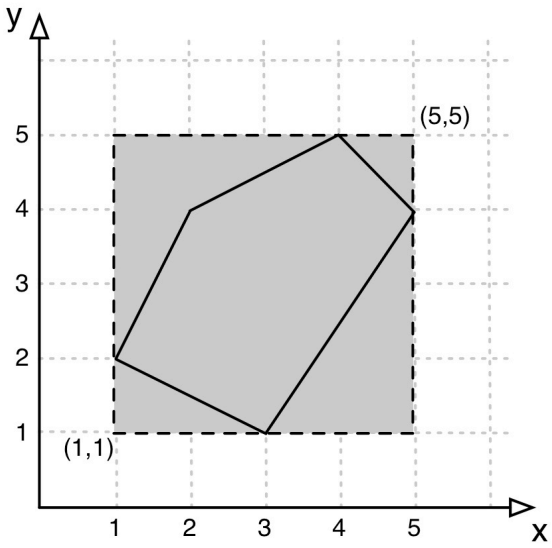
Oppgave 2 b) (8 %)
Omkretsen til et polygon er summen av kantlengdene i polygonet. Det finnes to spesialtilfeller. Et polygon med bare ett hjørnepunkt har omkrets lik 0, et polygon med to hjørnepunkter har omkrets lik 2 ganger kanten mellom punktene.

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 (*filename*) 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.

Om programmeringsløsninger i løsningsforslaget
I løsningsforslaget så returneres lister og tabeller som er endret ved hjelp av **return**. Dette er ikke nødvendig, men er gjort på denne måten av pedagogiske hensyn for å vise hva som returneres fra en funksjon.

Løsning Oppgave 2a)
```
def edgeLength(x1,y1,x2,y2):
    length = math.sqrt(math.pow(x1-x2,2)+(math.pow(y1-y2,2)))
    return length
```

Løsning Oppgave 2b)
```
def circumference(pList):
    N = len(pList) # Length of list
    res = 0
    if (N<2 or N%2<>0): # Test for odd number or less than 2
        res = -1
    elif N==2:
        res = 0
    elif N==4:
        res=2*edgeLength(pList[0],pList[1],pList[2],pList[3])
    else:
        for i in range(0,N-2,2):
            res = res +edgeLength(pList[i],pList[i+1],
                  pList[i+2],pList[i+3]) # should be one line
        res = res + edgeLength(pList[N-2],pList[N-1],
                  pList[0],pList[1]) # should be one line
    return res
```

Løsning Oppgave 2c)

```
def enclosingRectangle(pList):
    res = []
    xValues = pList[0::2] # List of all x-coordinates
    yValues = pList[1::2] # List of all y-coordinates

    res = [min(xValues),min(yValues),max(xValues),max(yValues)]
    return res
```

Alternativ løsning med for-løkke:

```
def enclosingRectangle(pList):
    N = len(pList)
    minx = pList[0]
    miny = pList[1]
    maxx = pList[0]
    maxy = pList[1]
    for i in range(2,N-1,2):
        if pList[i]>maxx:
            maxx = pList[i]
        if pList[i]<minx:
            minx = pList[i]
        if pList[i+1]>maxy:
            maxy = pList[i+1]
        if pList[i+1]<miny:
            miny = pList[i+1]
    res = [minx,miny,maxx,maxy]
    return res
```

Løsning Oppgave 2d)

```
def readPolygonFile(filename):
    res = []
    f = open(filename,'r')
    stringlist = f.readlines()
    f.close()
    for string in stringlist:
        words=string.split()
        for word in words:
            res.append(int(word))
    return res
```

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?

```
def mystery5(a,b,n):
    res = 0
    x = 0
    while (n>0):
        if (a[x]==b[x]):
            res = res + 1
        x = x + 1
        n = n - 1
    return res
```

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?

```
def mystery6(n,x):
    res = 0
    if n>0:
        res = x * mystery6(n-1,x)
    else:
        res = 1
    return res
```

Løsning Oppgave 3a)

Det returneres: 2

Løsning Oppgave 3b)

Funksjonen returnerer antall like elementer , i samme posisjon i de n første posisjonene i a og b.

Løsning Oppgave 3c)

Det returneres: 8

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 poengsummer og navn. Highscore-lista skal representeres som en dictionary der nøkkelen er plasseringen (1 til 10), og verdien er ei liste som består av navn og poengsum. Koden under viser et eksempel på opprettelse av en highscore-liste:

```
highscores = {}
highscores[1] = ['Vernon',100]
highscores[2] = ['Sirius',90]
highscores[3] = ['Severus',80]
…
highscores[10] = ['Albus',10]
```

Bruk funksjoner som defineres i andre deloppgavene 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, navn og poengsum som vist under. Merk at plassering skal skrives ut høyrejustert med 2 tegns feltbredde, navn venstrejustert med 20 tegns feltbredde, og poeng høyrejustert med 5 tegn feltbredde.

```
 1 Albus                100
 2 Frank                 90
 3 Fleur                 80
 4 Sirius                70
 5 Vernon                60
 6 Ron                   50
 7 Harry                 40
 8 Minerva               30
 9 Hermine               20
10 Severus               10
```

Oppgave 4 c) (10%)
Skriv funksjonen add_highscore som tar inn en poengsum (`points`) og et navn (`name`) og en highscore-liste (`scores`), og legger til poengsum og navn 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-listen som kan være enten uendret eller endret .

Figuren under viser highscore-lista før og etter add_highscore(65,'Luna',highscores) er kjørt:

Før:

1	Albus	100
2	Fleur	90
3	Frank	80
4	Harry	70
5	Hermine	60
6	Minerva	50
7	Ron	40
8	Severus	30
9	Sirius	20
10	Vernon	10

`add_highscore(65,'Luna',highscores)`

Etter:

1	Albus	100
2	Fleur	90
3	Frank	80
4	Harry	70
5	Luna	65
6	Hermine	60
7	Minerva	50
8	Ron	40
9	Severus	30
10	Sirius	20

Oppgave 4 d) (10%)
Skriv funksjonen most_highscores som tar inn en highscore-liste (`scores`) og returnerer navnet på den person som har flest innslag på lista. Hvis det er flere med like mange innslag, skal funksjonen returnere navnet til spilleren som er lengst oppe på lista. Hvis lista inneholder kun 10 forskjellige navn, skal en tom streng returneres.

Oppgave 4 e) (10%)
Skriv funksjonen new_highscorelist som returnerer en ny highscore-liste (dictionary) med poengsummer fra 100 ned til 10 (100, 90, 80…) der følgende ti navn skal plasseres tilfeldig i highscore-lista (merk at alle navn skal representeres i lista): Albus, Fleur, Frank, Harry, Hermine, Minerva, Ron, Severus, Sirius, og Vernon. Highscore-lista skal ha samme struktur som dictionary beskrevet i introduksjonen for oppgave 4.

Løsning Oppgave 4a)

```
def check_highscore(points, scores):
    for place in scores:
        if points>scores[place][1]:
            return place
return -1
```

Løsning Oppgave 4b)

```
def print_highscores(scores):
    for x in scores:
        print str(x).rjust(2),scores[x][0].ljust(20),
              str(scores[x][1]).rjust(5) # should be one line
```

Løsning Oppgave 4c)

```
def add_highscore(points,name,scores):
    place = check_highscore(points, scores)
    if place <> -1:
        for x in range(10,place,-1): # Move lower scores down
            scores[x] = scores[x-1]
        scores[place] =[name,points]
    return scores
```

Løsning Oppgave 4d)

```
def most_highscores(scores):
    entries = 1
    name = ''
    for x in scores:
        count = 0
        for i in scores:
            if scores[x][0]==scores[i][0]:
                count = count + 1
        if (count>entries):
            entries = count
            name = scores[x][0]
    return name
```

Løsning Oppgave 4e)

```
def new_highscorelist():
    import random
    scores = {}
    namelist = ['Albus','Fleur','Frank','Harry','Hermine',
      'Minerva','Ron','Severus','Sirius','Vernon'] # One line
    point = 100
    x = 1
    while len(namelist)>0:
        pick = int(random.random()*len(namelist))
        pickedname = namelist.pop(pick)
        scores[x] = [pickedname,point]
        point = point - 10
        x = x + 1
    return scores
```

Svarskjema flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

Oppgavenr	*A*	*B*	*C*	*D*
1.1	O			
1.2	O			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	