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 - JSP 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.

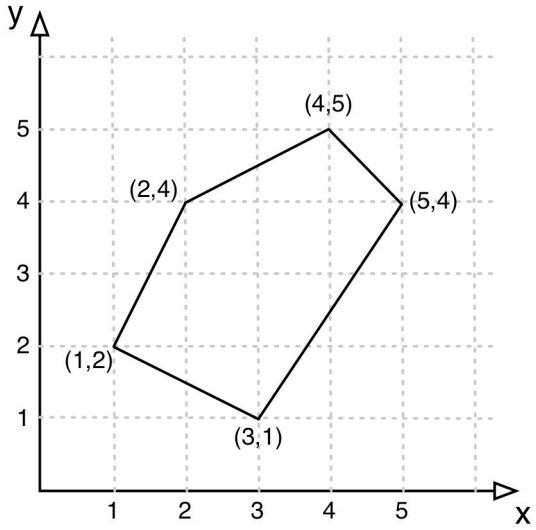
18) Hva er en algoritme?
a) Krav som stilles til et dataprogram.
b) En test for å finne feil i et dataprogram.
c) En presis beskrivelse av operasjoner som skal utføres for å løse et problem.
d) Ingen av alternativene er riktig.

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 metode `edgeLength` som tar inn koordinatene til to punkter som parametere (*heltall*) og som returnerer lengden av kanten mellom punktene (*desimaltall*).

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

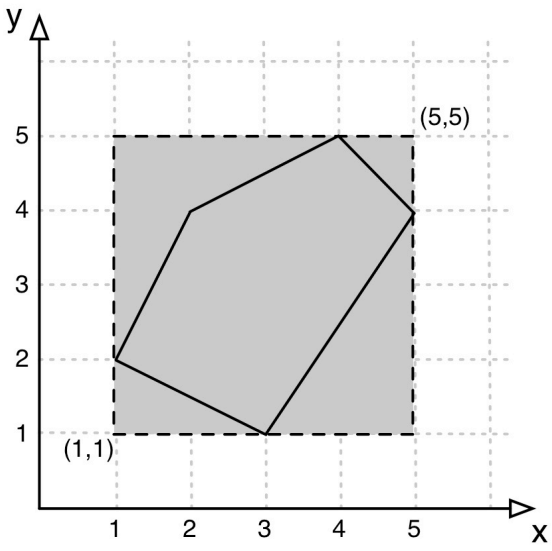
Oppgave 2 b) (8 %)
Omkretsen til et polygon er summen av kantlengdene i polygonet. Det finnes to spesialtilfeller. Et polygon med bare ett hjørnepunkt har omkrets lik 0, et polygon med to hjørnepunkter har omkrets lik 2 ganger kanten mellom punktene.

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

Hvis metoden kalles opp med `A` der `int[] A={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 metoden `edgeLength` fra deloppgave a. Du kan bruke denne metoden 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

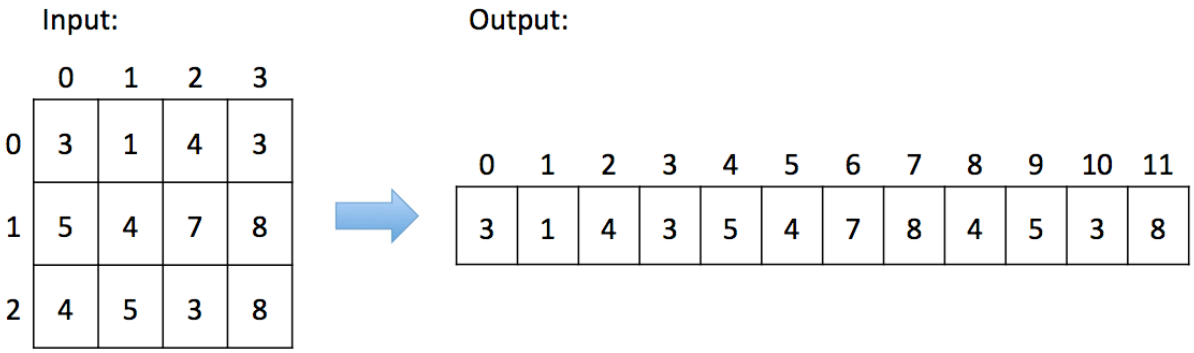
Det omsluttende rektangelet representeres med koordinatene til det nedre, venstre hjørnepunktet og koordinatene til det øvre, høyre hjørnepunktet.

Lag en metode `enclosingRectangle` som tar inn `pList` som er en liste av *heltall* 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 (liste av *heltall)*.

Dersom metoden kalles opp med `A` der `int[] A={3,1,5,4,4,5,2,4,1,2};` (femkanten i figur 4), skal metoden returnere en tabell med verdiene `{1,1,5,5}`.

Oppgave 2 d) (8 %)
Skriv en metode `getFromTable` som tar inn en todimensjonal tabell med *heltall* (`table`) av vilkårlig størrelse og legger x og y koordinater fra `table` inn i en endimensjonal tabell av *heltall* som skal returneres.

Hvis man kaller metoden `getFromTable` med parameteren `B`, der
`int[][] B = {{3,1,4,3},{5,4,7,8},{4,5,3,8}};`
skal metoden returnere en tabell med verdiene `{3,1,4,3,5,4,7,8,4,5,3,8}`.
Se illustrasjon under:



Vi vet ikke størrelsen på tabellen annet enn at antall elementer i x-aksen alltid er et partall.

Løsning 2a)
```
double edgeLength(int x1,int y1,int x2,int y2) {
      double res = Math.sqrt(Math.pow(x1-x2,2)+Math.pow(y1-y2,2));
      return res;
}
```

Løsning 2b)
```
double circumference(int[] pList) {
      int N = pList.length;
      double res = 0;
      if (N==2) {
            res = 0;
      } else if (N==4) {
            res = 2*edgeLength(pList[0],pList[1],pList[2],pList[3]);
      } else {
            for (int i=0;i<N-2;i=i+2) {
                  res=res+
                  edgeLength(pList[i],pList[i+1],pList[i+2],pList[i+3]);
            }
            res=res+edgeLength(pList[N-2],pList[N-1],pList[0],pList[1]);
      }
      return res;
}
```

Løsning 2c)

```
int[] enclosingRectangle(int[] pList) {
      int N = pList.length;
      int minx = pList[0];
      int miny = pList[1];
      int maxx = pList[0];
      int maxy = pList[1];
      for (int i=0;i<N-1;i=i+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];
            }
      }
      int[] res= {minx,miny,maxx,maxy};
      return res;
}
```

Løsning 2d)

```
int[] getFromTable(int[][] table) {
      int ylength = table.length;
      int xlength = table[0].length;
      int[] result = new int[xlength*ylength];
      int i = 0;
      for (int y=0;y<ylength;y=y+1) {
            for (int x=0;x<xlength;x=x+2) {
                  result[i]=table[y][x];
                  result[i+1]=table[y][x+1];
                  i=i+2;
            }
      }
      return result;
}
```

Oppgave 3 – Kodeforståelse (10%)

Oppgave 3 a) (4 %)
Hva returneres hvis metoden `mystery5(A,B,4)` med kode som vist under kjøres med følgende verdier for A og B:

```
int[] A = {3,4,6,7};
int[] B = {3,4,7,6};


int mystery4(int[] a, int[] b, int n) {
     int res = 0;
     int 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 metoden `mystery5` gjør.

Oppgave 3 c) (3 %)
Hva returneres hvis metoden `mystery6(3,2)` med kode som vist under kjøres?

```
int mystery6(int n, int x) {
     int res = 0;
     if (n>0) {
          res = x * mystery6(n-1,x);
     } else {
          res = 1;
     }
     return res;
}
```

Løsning 3a)
Det returneres: 2

Løsning 3b)
Metoden returnerer antall like elementer i samme posisjon i de n første posisjonene i a og b.

Løsning 3c)
Det returneres: 8

Oppgave 4 – Programmering (40 %)

I denne oppgaven skal du programmere ulike metoder 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
0	22048700	100
1	23313050	90
2	73595000	80
3	22000000	70
4	23048000	60
5	81544000	50
6	73594485	40
7	73590770	30
8	73593676	20
9	73591839	10

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

Oppgave 4 a) (5 %)

Skriv metode `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. Anta at poengsum angis som et heltall og at highscore-lista er en tabell av heltall.

Oppgave 4 b) (5 %)

Skriv JSP-koden for å skrive ut alle highscores med plassering, telefonnummer og poengsum som vist under (skriv i JSP-script og ikke som egen metode):

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 metoden `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. Metoden skal returnere highscore-lista som kan være enten uendret eller endret. Anta at poengsum og telefonnummer er av typen *heltall* og at highscore-lista er en tabell av *heltall*.

Figuren under viser highscore-lista før og etter at `add_highscore(65,90909090,highscores)` er kjørt:

Før:

0	22048700	100
1	23313050	90
2	73595000	80
3	22000000	70
4	23048000	60
5	81544000	50
6	73594485	40
7	73590770	30
8	73593676	20
9	73591839	10

`add_highscore(65,90909090,highscore)`

Etter:

0	22048700	100
1	23313050	90
2	73595000	80
3	22000000	70
4	90909090	65
5	23048000	60
6	81544000	50
7	73594485	40
8	73590770	30
9	73593676	20

Oppgave 4 d) (10 %)
Skriv metoden `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 metoden returnere telefonnummeret til den av spillerne som er lengst oppe på lista. Hvis lista inneholder kun 10 forskjellige nummer, skal tallet 0 returneres. Anta at highscore-liste er en tabell av *heltall*.

Oppgave 4 e) (10 %)
Skriv metoden `new_highscorelist` som returnerer en ny highscore-liste (todimensjonal tabell av *heltall*) 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.

Løsning 4a)

```
int check_highscore(int score, int[][] highscores) {
      int place=-1;
      for (int i=0;i<highscores.length;i++) {
            if (score>highscores[i][1]) {
                  place = i;
                  break;
            }
      }
      return place;
}
```

Løsning 4b)

```
for (int i=0;i<highscores.length;i++){
      out.println(i+1+" "+highscores[i][0]+ " "+ highscores[i][1]+"<br>");
}
```

Løsning 4c)

```
void add_highscore(int score, int phonenr, int[][] highscores) {
      int place = check_highscore(score,highscores);
      if (place!=-1) {
            for (int i=9;i>place;i--) {
                  highscores[i][0] = highscores[i-1][0]; // Flytter nummer
                  highscores[i][1] = highscores[i-1][1]; // Flytter score
                  }
            highscores[place][0] = phonenr;
            highscores[place][1] = score;
      }
}
```

Løsning 4d)

```
int most_highscores(int[][] highscores) {
      int entries = 1;
      int number = 0;
      int count = 0;
      for (int i=0;i<highscores.length;i++) {
            for (int j=0;i<highscores.length;j++) {
                  if (highscores[i][0]==highscores[j][0]) {
                        count = count + 1;
                  }
            }
            if (count>entries) {
                  entries = count;
                  number = highscores[i][0];
            }
      }
      return number;
}
```

Løsning 4e)

```
int[][] new_highscorelist() {
      int[][] highscores = new int[10][10];
      int[] numbers = {73594400, 73594485, 22000000, 90909090, 73500000,
      73000000, 22220000, 54000000, 30303030, 40404040};
      int points = 100;
      int x = 0;
      while (x<10) {
            int place = (int) (Math.random()*10);
            if (numbers[place]!=0) {
                  highscores[x][0]= numbers[place];
                  highscores[x][1]= points;
                  numbers[place]=0;
                  points = points - 10;
                  x=x+1;
            }
      }
      return highscores;
}
```

Svarskjema flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

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