

Jon Arve Ovesen 770215 T015
a06jonov@his.se

Innleveringsoppgave 1

Procedurell programmering

Mastermind

Analyse

Design

Implementasjon

Test

Analyse betyr å bryte ned et kompleks problem ned i mindre, og mer håndterlige uavhengige deler i den hensikt å inspesere noe nøye, og med håp om at å løse disse mindre delene også vil føre til å løsning for det mer komplekse problemet. I mitt tilfelle med spillet Mastermind, var det første jeg måtte gjøre, å finne ut så mye om spillet som mulig, spesielt reglene. Dette stod en del i oppgaven men jeg brukte også forskjellige oppslagsverk på internet.

Mastermind er et simpelt kode-løsnings brett spill for to spillere, og ble oppfunnet av Mordecai Meirowitz som er en israelsk telekommunikasjons ekspert og er en videre utvilking av spillet Bulls and Cows som ble spilt med penn og papir. Kort forklart går spillet ut på å gjette fire pinner farge og sekvens. Der spiller A i all hemlighet velger fire farger og deres sekvens. Det er da opp til den andre spilleren B å prøve og gjenskape den både de riktige fargene og deres plassering (sekvens) med et antall gjettinger. Spiller A kommer så med feedback med 0 til 4 pinner. En farget pinne (eller svart) blir satt opp hvis en av pinnene har både riktig farge og plassering. Og en hvit pinne blir satt opp hvis det er satt opp riktig farge men på feil plassering. Spiller A kan så gjette pånytt og har tilsammen 12 forsøk (noen ganger 10 eller 8) på å gjette riktig kode.

Det andre kom så til å planlegge hvordan dette skulle programmeres, og dele opp det opp i små håndterlige deler som jeg kunne skrive C++ kode for. Et enkelt valg jeg tar for å gjøre dette lettere er å bytte ut fargene med representative tall for fargene. Og noe av det mest innlysende er at jeg trenger to grunnleggende variabler i spillet. Den hemlige kodens variabel og spillerens gjettings variabel. Siden jeg velger tall for å representere fargene kan jeg bruke array av typen int. og slipper da unna med 2 variabler istedet for å bruke 4 variabler for hver som ellers ville blitt 8 tilsammen. Et annet arument er at det er at det muligens er lettere for spilleren å taste fire tall enn å skrive fire farger (Dette kan kanskje testes ut etterhvert.). Siden jeg har tenkt å kjøre og teste spillet hjemme på et windows basert miljø, har jeg laget spillet til dette formålet.

De små delproblemene deler jeg så inn i følgende deler:

1. Main del med krav om mulighet for bruk av øæå.
2. Meny del der kravet for koden er at den skal gi brukeren mulighet til å velge fra en meny (for senere implementasjon av flere spill)
3. Hoveddel - som kalles mastermind () med krav om at koden skal gi en omgangs konkusjon , samtidig telle antall omganger og inneholde spillets gang (rekkefølge på de forskjellige delene).
4. Intro del der kravet for koden er at den skal forklare litt om spillets regler og at spilleren kan velge om det skal spilles den forenklete versjonen eller den vanlige versjonen.
5. Random del for forenklet versjon, der kravet er at koden skal kunne plassere fire tilfeldige tall som er unike i et array. (Deloppgave a)
6. Random del for den vanlige versjonen. der kravet er at koden skal kunne plassere fire tilfelige tall i et array (ikke nødvendigvis unike). (Deloppgave b)
7. Inntastings del der kravet er at spilleren kan taste inn sin gjetting av koden, og at det blir lagret i et array.
8. Feilmeldings del - der kravet er at koden skal kunne gi feedback på feil inntasting.
9. Svart pinne del - der kravet er at koden skal kunne sjekke hvor mange riktige tall det er på riktig plass.
10. Hvit pinne del - der kravet er at koden skal kunne sjekke hvor mange riktige tall det er på feil plassering.

11. Hoveddel - med krav om at koden skal gi en omgangs konkusjon telle antall omganger og inneholde spillets gang (rekkefølge på de forskjellige delene).
12. Outro del - med krav om at koden skal gi spørsmål om spilleren vil spille en gang til og eventuelt starte spillet pånytt.

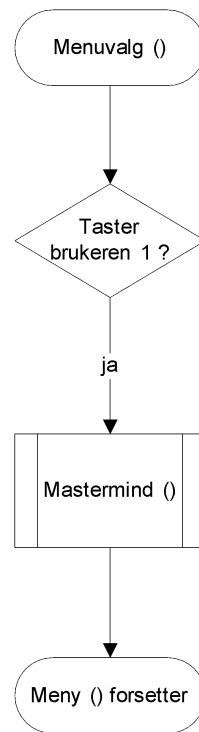
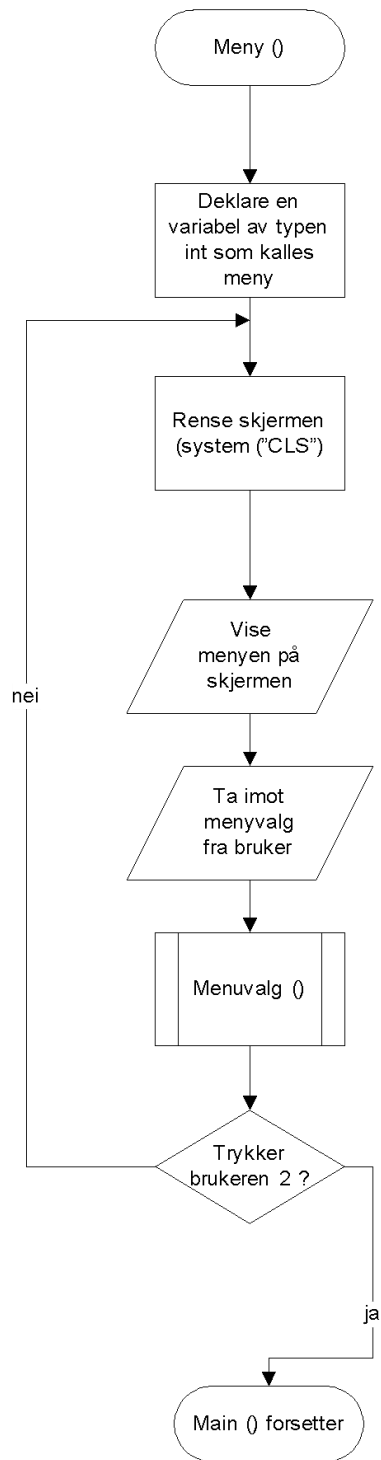
Å designe refererer som verb til prosessen å produsere en plan for et produkt, stuktur, system eller komponent. Og som substantiv blir "et design" brukt for enten den ferdige løsningen (planen) som f.eks et forslag, tegning, modell eller diskripsjon, eller også resultatet av å implemenetere denne løsningen.

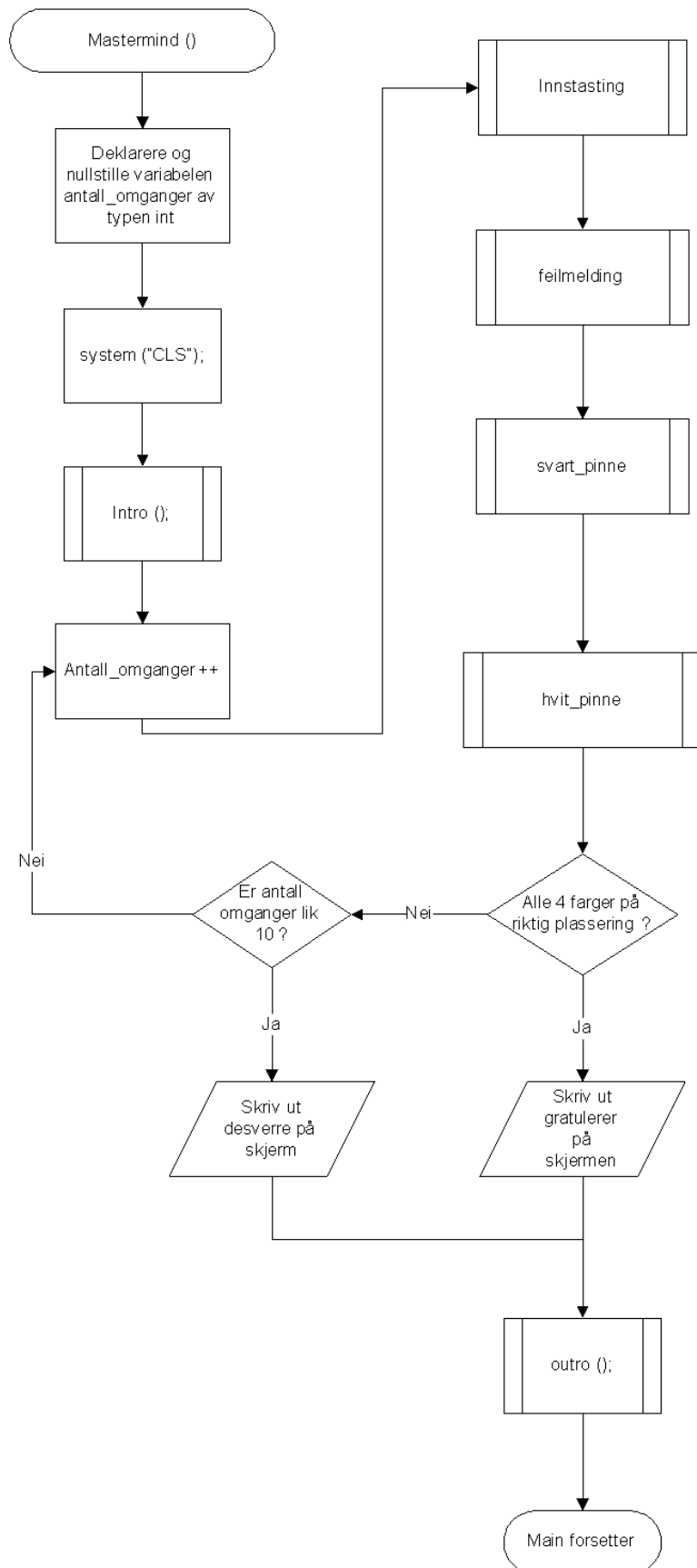
Design og analyse glir ofte inn i hverandre og jeg har vel allerede nevnt ting som inngår i design fasen. I oppgaven blir vi satt til å designe programmerings oppgaven med et **flytskjema** til den ene deloppgaven og et **JSP diagram** for den andre. Jeg har derimot valgt og lage begge versjone i et og samme spill og gir derfor både et detaljert flytskjema og JSP diagram for delen mastermind () for denne ene løsningen.

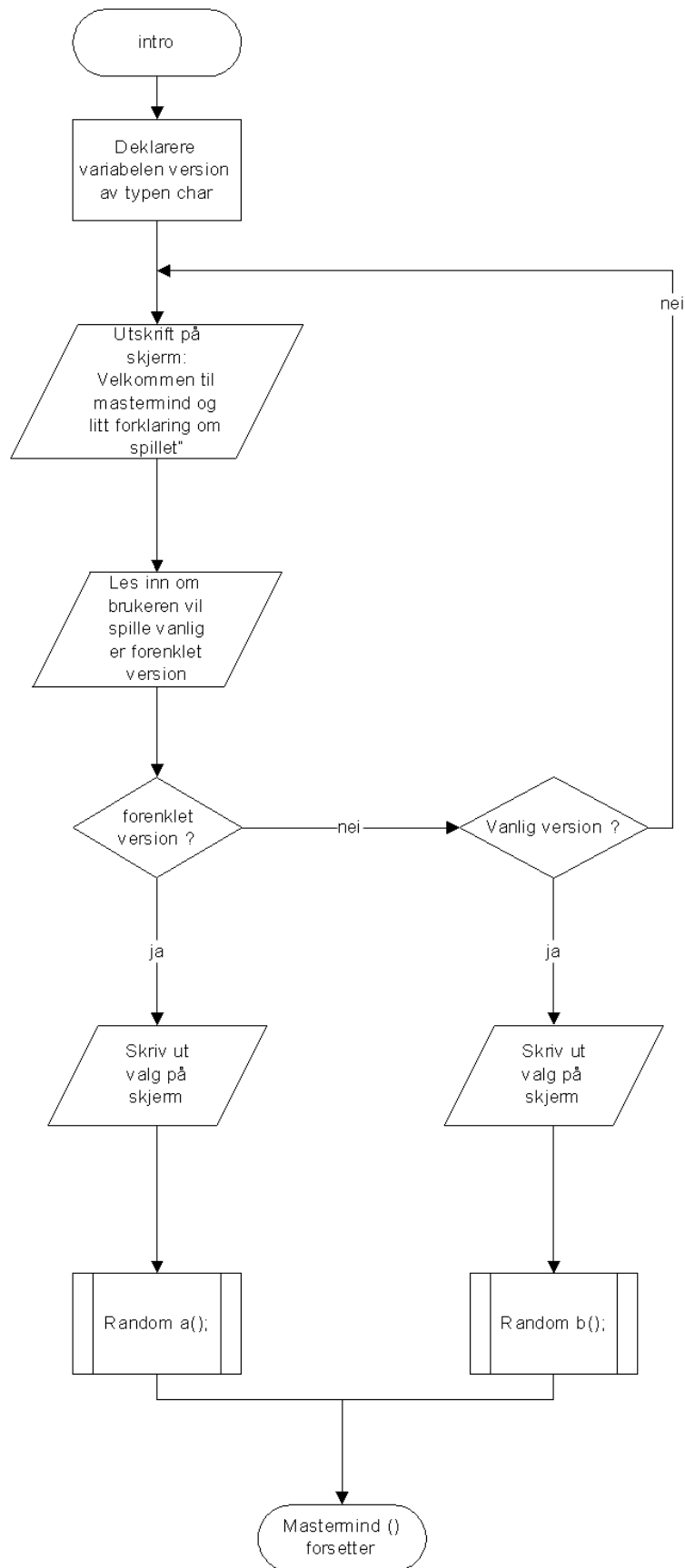
Å lage flyt skjema gikk greit og jeg brukte Microsoft Office Visio til å lage disse. Jeg laget et flyskjema for hver del (funksjon eller scope) av koden.

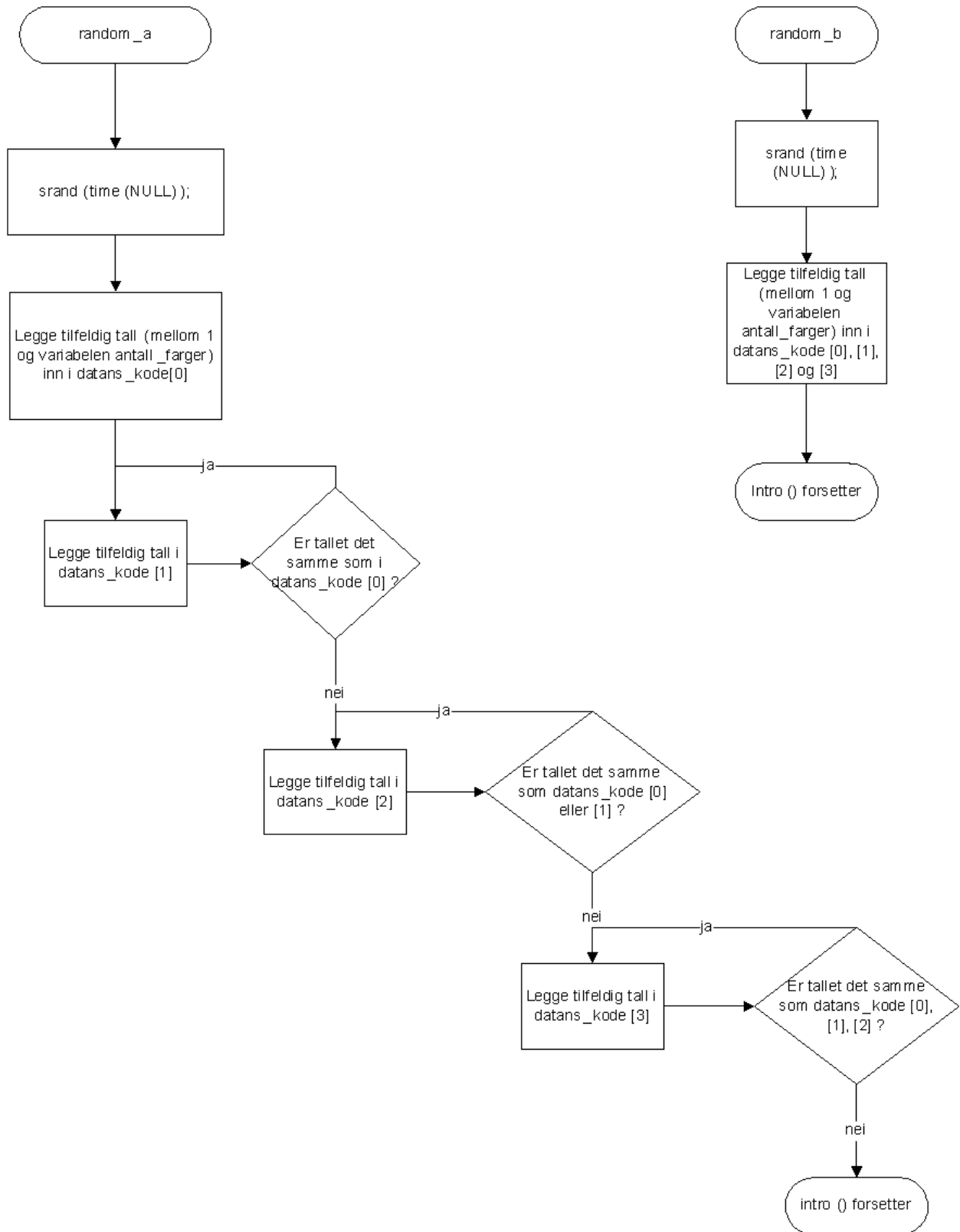
Her kommer flytskjema over de respektive delene:

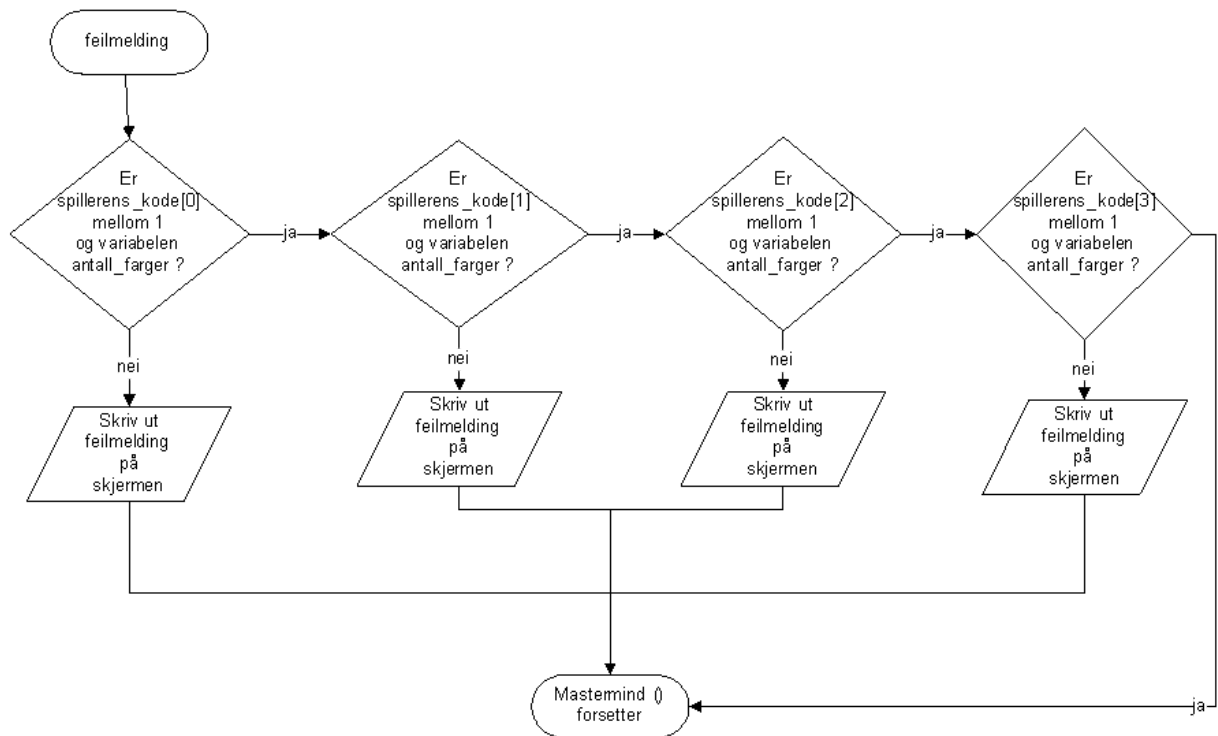
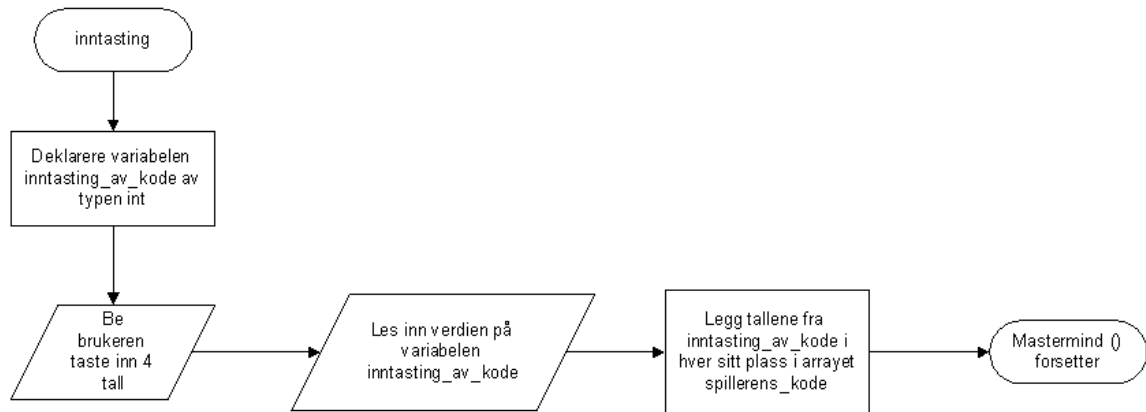


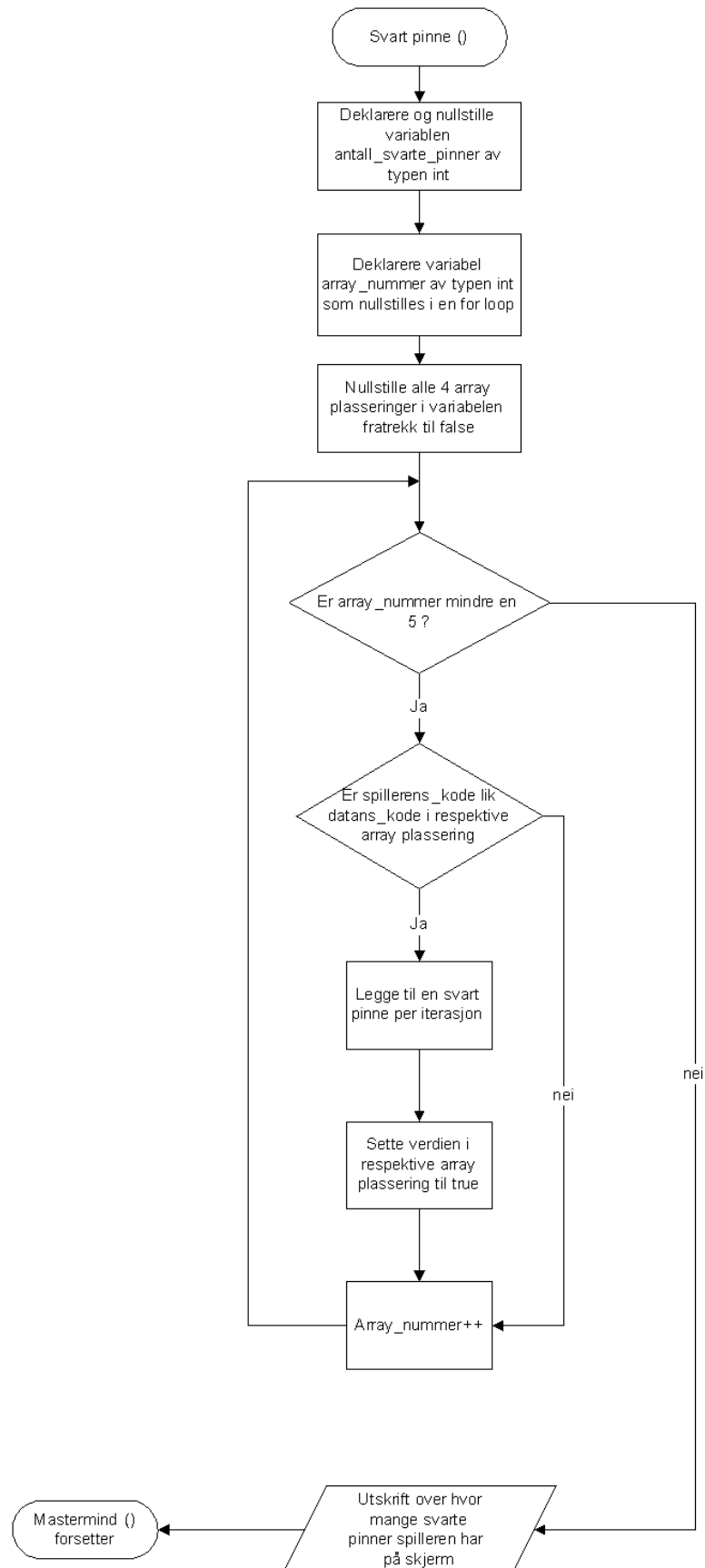


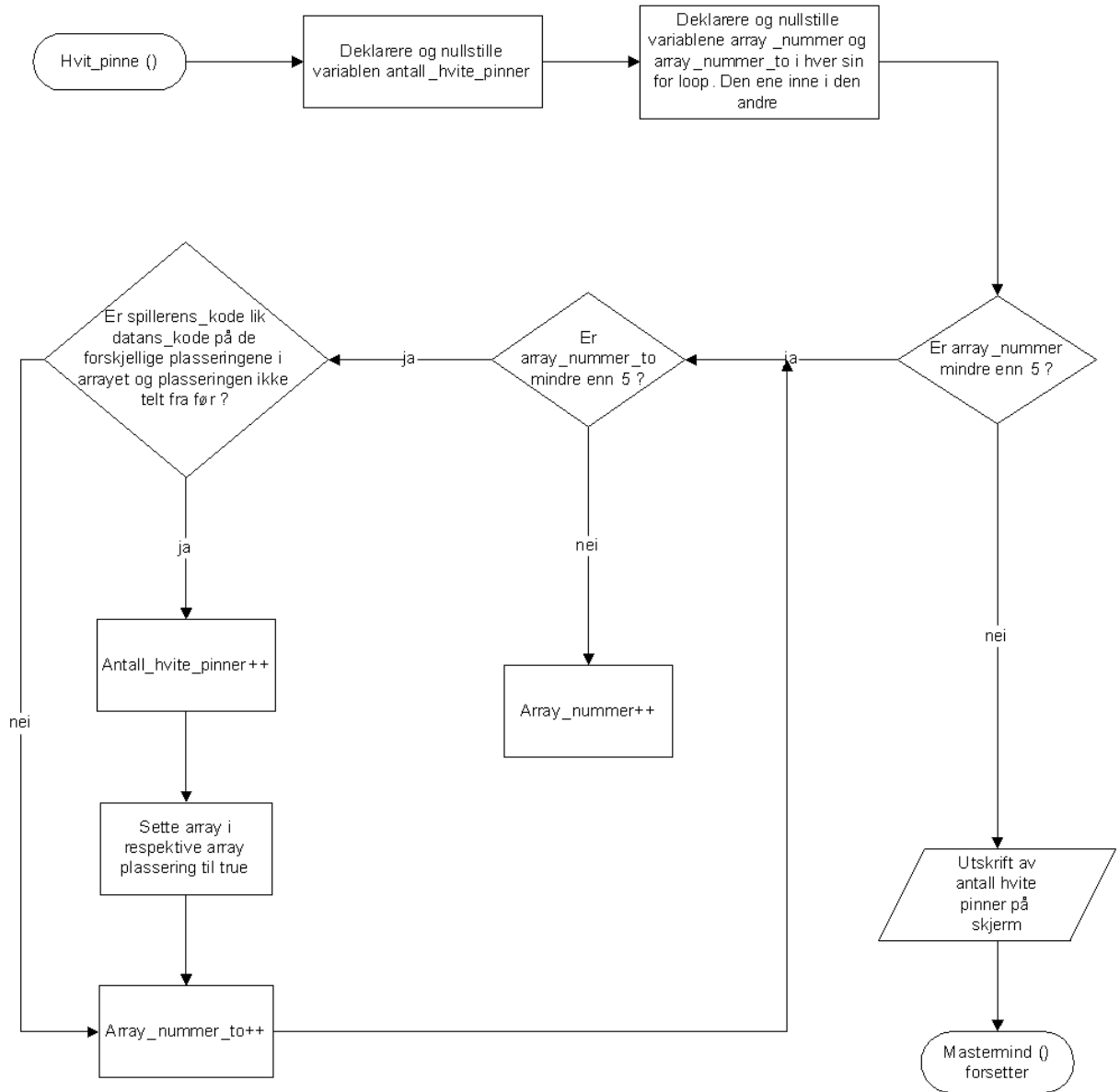




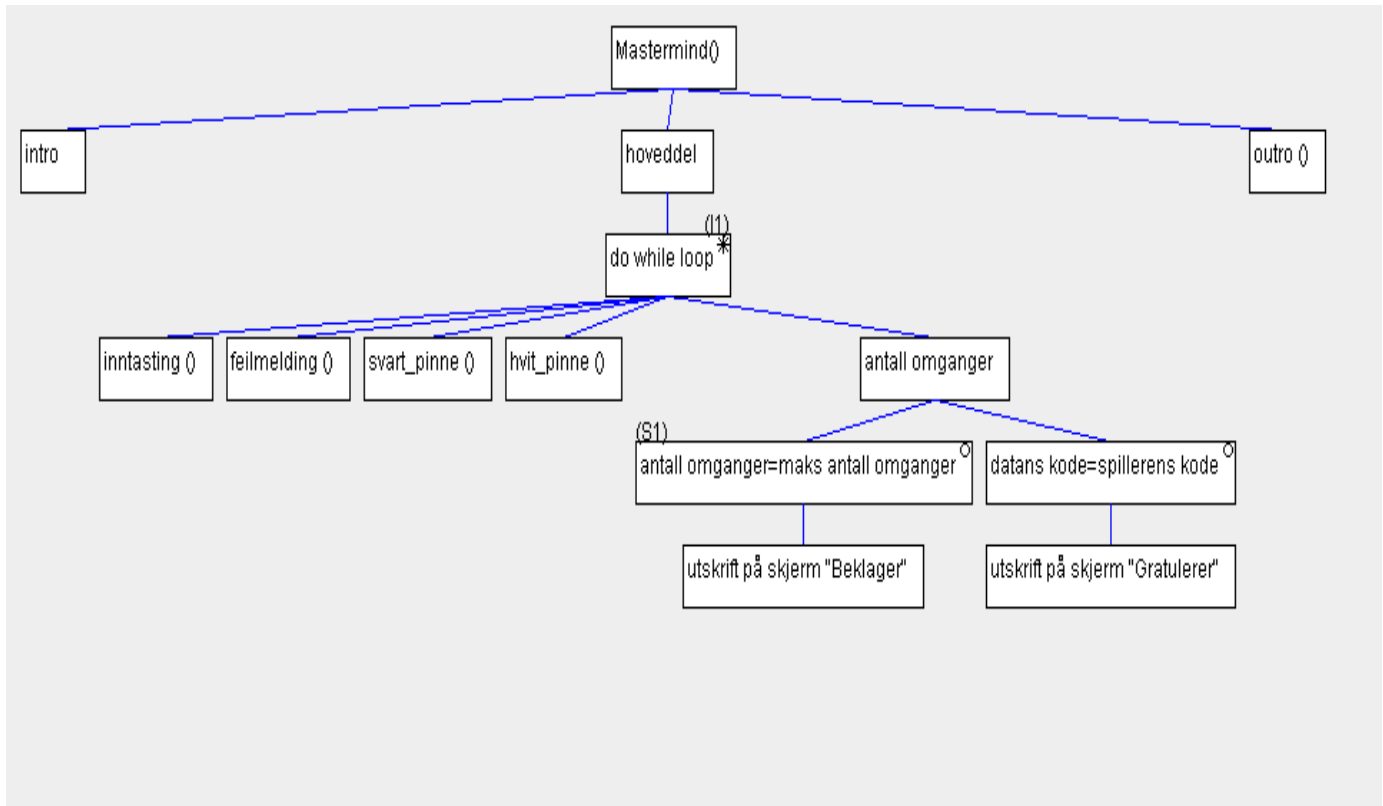








Her er et Jsp diagram over delen Mastermind () som vi ble enige om at holdt siden jeg laget 2 in 1. Selv om jeg brukte JSP diagram for mange av de andre delene også, da man lett kan genere kode av et slikt diagram. Denne er laget med JSP editor 3.0 av Henrik Engstrøm.



Vilkårsliste:

Iterasjon 1 = datans_kode!=spillerens_kode
 Seleksjon 1 = antall_omganger==maks_antall_omganger
 Seleksjon 2 = datans_kode==spillerens_kode

Koden som ble generert av dette JSP diagrammet så slik ut:

```
int mastermind()

{
    /* intro */

    while (datans_kode!=spillerens_kode)
    {
        /* inntasting () */

        /* feilmelding () */

        /* svart_pinne () */

        /* hvit_pinne () */

        if (antall_omganger==maks_antall_omganger)

            /* utskrift på skjerm "Beklager" */

        else

            /* utskrift på skjerm "Gratulerer" */

        }

        /* outro () */

    }
```

Implimentasjon betyr det å gjennomføre, fullføre eller hvordan en plan faktisk skjer. Som følger er altså implementasjon en handling som må følge etter det innledende skrittet for at noe faktisk skal skje. Og i denne sammenhengen det å skrive en C++ kode utifra de foregående opplysninger. Her kommer den:

```
/*Denne C++ koden kallt "Masterminded" er laget med Visual C++ Studio 2008
Express Edition.

Spillet går ut på å gjette de fire pinnenenes farge og sekvens.
Fargene er representert av tall.

Svart pinne betyr at det er rett tall på riktig plass.
Hvit pinne betyr at du har rett tall men på feil plass.

a06jonov@his.se

*/

#include <iostream> /*Inkludering av iostream siden jeg skal bruke funksjonene
cin (istream), og cout (ostream).*/
#include <stdlib.h> /*Inkludering av stdlib header siden jeg skal bruke srand
funksjonen som finnes i denne. Se
http://www.cplusplus.com/reference/clibrary/cstdlib/srand.html for mer info. */
#include <time.h> /*Inkludering av header file time siden jeg skal bruke
klokken i systemet til random pseudo generator.
Se http://www-ccs.ucsd.edu/c/time.html om denne.*/
#include "iodos.h" /*Inkludering av header file iodoss, siden jeg skal vil
bruke øå i dos consol.*/
#define antall_farger 6/*Her defineres antall farger som må være over 4.*/
#define maks_antall_omganger 10 /*Her defineres antall forsøk spilleren har på
å gjette riktig kode.*/
#define false 0 /*definerer 0 som falsk.*/
#define true 1 /*definerer 1 som sant.*/

using namespace std; /*Siden alle filene i standardiserte C++ biblioteket er
deklarerer i std namespace, slipper jeg unna med å benevne denne deklarasjonen
uten å spesifisere nærmere. */

/*Disse "globale" og selvforklarende variablene som her blir deklarerer kan
fjernes ved å putte "funksjonene" sammen og ha de i en og samme del.
Men siden jeg brukte Visual Studio så forenklet det arbeidet mye og gjorde det
ryddigere at jeg kunne hoppe til de forskjellige funksjonene med Global Scope
*/
int datans_kode[4], spillerens_kode[4], telt_plassering[4];
int mastermind();

/*Her er "funksjonen" som finner antall hvite pinner i spillet*/
```

```

void hvit_pinne()
{
    int antall_hvite_pinner=0; /*Deklarering og nullstilling av variabel for
telling av antall hvite pinner */

    /*Hvis spillerens_kode er lik datans_kode på andre plasseringer enn den
samme og plasseringen
ikke er telt fra før, så blir det en hvit pinne: */

    for (int i=0; i<4; i++) /*Her deklarerer i av typen int samtidig med for
loopen */
    {
        for (int j=0; j<4; j++) /*Her deklarerer j av typen int samtidig
med for loopen */
        {
            if ((spillerens_kode[i]==datans_kode[j]) &&
telt_plassering[j]==false)
            {
                antall_hvite_pinner++;
                telt_plassering[j]=true;
            }
        }
    }

    cout <<"\tDu har " <<antall_hvite_pinner <<" stk hvite pinner.\n";
}

/* Her er funksjonen som teller svarte pinner i spillet.*/
void svart_pinne()
{
    int antall_svarte_pinner=0; /*Deklarering og nullstilling av variabelen
som skal telle antall svarte pinner*/

    /* Nullstilling av telt_plassering til false på alle 4 plasseringer i
arrayet */
    for (int i=0; i<4; i++)
    {
        telt_plassering[i]=false;
    }

    /*Hvis spillerens_kode er lik datans_kode på en plassering i arrayet så blir
det en svart pinne: */

    for (int j=0; j<4; j++) /*Her deklarerer i av typen int samtidig med for
loopen */
    {
        if (spillerens_kode[j]==datans_kode[j])
        {
            antall_svarte_pinner++;
            telt_plassering[j]=true;
        }
    }
    cout <<"\tDu har " <<antall_svarte_pinner <<" stk svarte pinner.\n";
}

```

```

/*Dette er delen for inntasting av kode (gjetting av mastermind kode)*/
void inntasting()
{
    int inntasting_av_kode; /*Deklarering av variabel av typen int for
inntasting av gjetting.*/
    cout <<"\n\tGjett 4 tall mellom 1 og " <<antall_farger <<": ";
    cin >>inntasting_av_kode;

    /*Her deles de fire sifrede tallet opp til fire respektive tall
    som blir lagret på hver sin plassering i et array: */

    spillerens_kode[0]=int(inntasting_av_kode/1000);
    spillerens_kode[1]=(int(inntasting_av_kode/100))-spillerens_kode[0]*10;
    spillerens_kode[3]=inntasting_av_kode%10;
    spillerens_kode[2]=((inntasting_av_kode%100)-spillerens_kode[3])/10;

}

```

/* Denne feilmeldingsdelen for inntasting av kode betraktes som "Alpha" da jeg ikke er fornøyd med den, selv om den fungerer. */

```

void feilmelding ()
{
    if (spillerens_kode[0]<=0 || spillerens_kode[0]>antall_farger)
    {
        cout <<"\tDu tastet feil.\n";
    }
    else if (spillerens_kode[1]<=0 || spillerens_kode[1]>antall_farger)
    {
        cout <<"\tTallene du tastet var ikke mellom 1 og " <<antall_farger
<<"\n";
    }
    else if (spillerens_kode[2]<=0 || spillerens_kode[2]>antall_farger)
    {
        cout <<"\tTallene du tastet var ikke mellom 1 og " <<antall_farger
<<"\n";
    }
    else if (spillerens_kode[3]<=0 || spillerens_kode[3]>antall_farger)
    {
        cout <<"\tTallene du tastet var ikke mellom 1 og " <<antall_farger
<<"\n";
    }
}

```

```

/*Denne "funksjonen" eller "scopet" er basert på psudo random som legger fire
forskjellige tall i et array:*/

void random_a()
{
/*pseudo-random number generator som tar tilfeldig tall fra klokken til
systemet (random seed): */
    srand (time (NULL) );

/* Her legges det første tallet i arrayet plassering nr.0 */
    datans_kode[0] = rand()%antall_farger+1;

/* Her legges det andre tallet i arrayet plassering nr.1 og gjentas hvis tallet
er det samme som det første */
    do
    {
        datans_kode[1] = rand()%antall_farger+1;
    }
    while (datans_kode[1]==0 || datans_kode[1]==datans_kode[0]);

/* Her legges det tredje tallet i arrayet plassering nr.2 og gjentas hvis
tallet er det samme som noen av de andre */
    do
    {
        datans_kode[2] = rand()%antall_farger+1;
    }
    while (datans_kode[2]==0 || datans_kode[2]==datans_kode[1] ||
datans_kode[2]==datans_kode[0] );

/* Her legges det fjerde tallet i arrayet plassering nr.3 og gjentas hvis
tallet er det samme som noen av de andre */
    do
    {
        datans_kode[3] = rand()%antall_farger+1;
    }
    while (datans_kode[3]==0 || datans_kode[3]==datans_kode[2] ||
datans_kode[3]==datans_kode[1] || datans_kode[3]==datans_kode[0]);

}

/*Mens i denne "funksjonen" med psudo random generering er sjansen faktisk
ganske stor for at den blir flere av samme tall.*/
void random_b()
{
    srand (time (NULL) );
    datans_kode[0] = rand()%antall_farger+1;
    datans_kode[1] = rand()%antall_farger+1;
    datans_kode[2] = rand()%antall_farger+1;
    datans_kode[3] = rand()%antall_farger+1;

}

```



```

/*Avslutting av spillet hvis spilleren ikke taster inn j eller J, isåfall
startes delen Mastermind() pånytt: */
void outro ()
{
char ja_nei;

cout <<"\tVil du spille igjen (j/n) ? ";
cin >>ja_nei;

if (ja_nei=='j' || ja_nei=='J')
{
    mastermind();
}
else
{
    cout <<"\tHadet bra ! Velkommen tilbake.\n";
}
}

void intro()
{
    char version; /*Deklarering av variabelen version av typen char*/
    do
    {
        system ("CLS"); /*Renser skjermen*/
        cout <<"\tVelkommen til Mastermind !\n\n";
        cout <<"\tSpillet går ut på å gjette de fire pinnenes farge og
sekvens.\n";
        cout <<"\tFargene er represenert av tall mellom 1 og " <<antall_farger
<<".\n";
        cout <<"\tI forenklet versjon gjetter du på fire forskjellige tall.\n";
        cout <<"\tMens i vanlig versjon kan det være flere av det samme
tallet.\n\n";
        cout <<"\tSvart pinne betyr at du har rett tall på riktig plass.\n";
        cout <<"\tHvit pinne betyr at du har rett tall men på feil plass.\n\n";
        cout <<"\tVil du spille forenklet eller vanlig version ? \n\n";
        cout <<"\tTast f eller v etterfulgt av <enter>: ";
        cin >>version;

        if (version=='f' || version=='F')
        {
            cout <<"\n\tDu har valgt å spille den forenklede versionen.\n\n";
            random_a();
            break;
        }
        else if (version=='v' || version=='V')
        {
            cout <<"\n\tDu har valgt å spille den vanlige versionen.\n\n";
            random_b();
            break;
        }

    }while (version!='f' || version!='F' || version!='v' || version!='V' );

    cout <<"\tDatamaskinen har nå valgt fire tall.";

```

/*Her kommer selve hendelseforløpet til spillet. Dette er en modifisert utgave av den som ble generert av JSP-editoren (While loopen ble byttet ut med do while loop og else ble byttet ut med else if) */

```
int mastermind()
{
    int antall_omganger=0; //selvforklarende deklarerer av variabelen
    antall_omganger av typen int
    intro();

    do
    {
        inntasting();
        feilmelding();
        svart_pinne();
        hvit_pinne();

        /*Antall forsøk på å gjette riktig kode telles her. */
        antall_omganger++;

        /*Hvis maks antall forsøk blir nådd */
        if (antall_omganger==maks_antall_omganger &&
            (datans_kode[0]!=spillerens_kode[0] &&
            datans_kode[1]!=spillerens_kode[1] &&
            datans_kode[2]!=spillerens_kode[2] &&
            datans_kode[3]!=spillerens_kode[3]))
        {
            cout <<"\tDesverre, du klarte det ikke på "
                <<antall_omganger <<" forsøk.\n";
            cout <<"\tRiktig svar var: ";
                for (int i=0; i<4; i++)
                {
                    cout <<datans_kode[i];
                }
                cout <<endl; /*linjeskift*/

            break;
        }

        /*Hvis alle tallene og plassering er lik i
        datans_kode og spillerens_kode */
        else if (datans_kode[0]==spillerens_kode[0] &&
            datans_kode[1]==spillerens_kode[1] &&
            datans_kode[2]==spillerens_kode[2] &&
            datans_kode[3]==spillerens_kode[3])
        {
            cout <<"\n\tGratulerer du klarte det på "
                <<antall_omganger <<" forsøk\n";
        }

        /* disse linjene er bare for testing av programmet: */

        cout <<"\tDatan:      ";

        for (int i=0; i<4; i++)
        {
            cout <<datans_kode[i];
```

```

    }
    cout <<endl;

    cout <<"\tSpilleren:";

    for (int j=0; j<4; j++)
    {
        cout <<spillerens_kode[j];
    }
    cout <<endl;

    }while (datans_kode[0]!=spillerens_kode[0] ||
datans_kode[1]!=spillerens_kode[1] || datans_kode[2]!=spillerens_kode[2]
|| datans_kode[3]!=spillerens_kode[3]);

    outro();
    return 0;
}

/*Her kommer en meny som senere kan bygges ut med flere spill.
Siden jeg også har laget Stein, Saks, Papir og Gissa et tall: */
void menuvalg(int menu)
{
    switch (menu){
    default:break;
    case 1: mastermind();break;
    }
}

void meny()
{
    int menu;
    do
    {
        system ("cls");
        cout << "\n\t Velg fra menyen:\n";
        cout <<"\n\t1. Mastermind.\n";
        cout <<"\n\t2. Avslutte applikasjonen.\n\n";
        cin >> menu;
        cin.ignore(80, '\n');
        menuvalg(menu);
    }
    while(menu!=2);
    cout <<"\n\t";
}

int main()
{
    /* For øääå*/
    dos_console();

    meny();

    return 0;
}

```

Testing ble foretatt underveis og gikk hånd i hånd med implementasjonen. Etter at en del var ferdig skrive, testet jeg den og rettet opp små kompilerings feil. Det oppstod ikke så mange i den ganske enkle oppgaven men det oppstod noen typiske skrivefeil.

Det viste seg dog at under det 10.ende forsøket endte med "Desverre du klarte det ikke" selv om dette forsøket var riktig.

Koden var slik:

```
if (antall_omganger==maks_antall_omganger)
{
cout <<"\tDesverre, du klarte det ikke på "
  <<antall_omganger <<" forsøk.\n";
cout <<"\tRiktig svar var: ";
    for (int i=0; i<4; i++)
    {
        cout <<datans_kode[i];
    }
    cout <<endl; /*linjeskift*/

break;
}
```

og måtte rettes til slik:

```
if ( antall_omganger==maks_antall_omganger &&
( datans_kode[0]!=spillerens_kode[0] &&
  datans_kode[1]!=spillerens_kode[1] &&
  datans_kode[2]!=spillerens_kode[2] &&
  datans_kode[3]!=spillerens_kode[3] ))
{
cout <<"\tDesverre, du klarte det ikke på "
  <<antall_omganger <<" forsøk.\n";
cout <<"\tRiktig svar var: ";
    for (int i=0; i<4; i++)
    {
        cout <<datans_kode[i];
    }
    cout <<endl; /*linjeskift*/

break;
}
```

Jeg har aldri skrevet en slik rapport som dette før og viste ikke helt hvordan jeg skulle skrive den, men den ble da skrevet på et vis. Håper du er fornøyd ellers leverer jeg jo i god tid så jeg har sjans til å gjøre den bedre i tide.

Vennlig hilsen

Jon Arve Ovesen