

SENSORVEILEDNING

Emnekode:	IRE12011
Emnenavn:	Programmering og mikrokontrollere
Eksamensform:	Skriftlig (digital via Inspera)
Dato:	10.12.2018
Faglærer(e):	Åge Torbjørn Johansen
Eventuelt:	Versjon 19.12.2018

Hver oppgave / deloppgave er gitt en maksimalpoengsum. Fortrinnsvis benyttes følgende prosentskala på hele besvarelsen:

Karakter	Poeng
A	90% og over
B	80% - 89%
C	60% - 79%
D	50% - 59%
E	40% - 49%
F	Under 40%

Der det av diverse grunner kan være problematisk å benytte ovenstående skala, kan en helhetsvurdering etter følgende beskrivelser benyttes.

Karakter	Betegnelse	Beskrivelse
A	Fremragende	Fremragende prestasjon der kandidaten har løst problemer som krever fantasi og innsikt. Besvarelsen viser at kandidaten fullt ut behersker både de begrepsmessige, regnetekniske og anvendelsesmessige delene av emnet. Fremstillingen er klar og presis med korrekt bruk av notasjon og fagterminologi. Noen få mindre feil eller blanke punkter kan tillates.
B	Meget god	Meget god prestasjon der kandidaten har løst problemer som går utover det rutinemessige, og som krever god oversikt over emnet. Besvarelsen viser meget god beherskelse av de sentrale teknikkene, begrepene og anvendelsene i kurset. Fremstillingen er klar og med stort sett riktig bruk av terminologi og notasjon.
C	God	Gjennomsnittlig prestasjon der kandidaten har løst oppgaver av middels vanskelighetsgrad fra de fleste deler av kurset. Besvarelsen viser god beherskelse av de sentrale teknikkene, begrepene og anvendelsene i kurset, men kandidaten har ikke i særlig grad klart å anvende sine ferdigheter og kunnskaper på oppgaver som går ut over det rutinemessige. Fremstillingen er grei å forstå, men kan ha en del formelle mangler.
D	Nokså god	Prestasjon under gjennomsnittet der kandidaten har løst eller kommet et stykke på vei med oppgaver fra

		flere sentrale deler av kurset. Besvarelsen viser kjennskap til de viktigste teknikkene, begrepene og anvendelsene i kurset, men kandidaten har vanskelig for å komme helt i mål selv på rutinepregede oppgaver. Fremstillingen er stort sett forståelig, men kan ha en god del formelle mangler.
E	Tilstrekkelig	Prestasjon som tilfredsstillende minimumskravene, men heller ikke mer. Besvarelsen viser at kandidaten har kjennskap til begreper, teknikker og anvendelser fra flere deler av kurset, og at han/hun til en viss grad kan bruke sine kunnskaper til å løse oppgaver. Fremstillingen er stort sett forståelig, men røper klare feil og misforståelser.
F	Ikke bestått	Prestasjon som ikke tilfredsstillende minimumskravene. Besvarelsen viser at kandidaten har manglende kjennskap til sentrale teknikker, begreper og anvendelser, eller manglende evne til å bruke sine kunnskaper til å løse oppgaver. Besvarelser som bare viser beherskelse av en avgrenset del av emnet, vil normalt havne i denne kategorien.

Oppgavene 2 (30p) og 3 (20p) med deloppgaver krever konkrete svar uten begrunnelse, som vurderes automatisk av eksamensplattformen *Inspira*.

Riktige svar vil presenteres for sensor når sensurering starter, men for en oversikt over alle spørsmål, vises disse til slutt i dette dokumentet.

Hensikten med disse spørsmålene er å la kandidaten vise faktakunnskap innen ulike områder når det gjelder programmeringsspråket C/C++.

Evnen til å lese programmer testes også.

NB! I poenggivingen for oppgave 3.1 er det en feil i det automatiske oppsettet. Denne deloppgaven bør derfor sensureres manuelt for de kandidater som ligger i nærheten av poenggrensen mellom to karakterer.

Løsningsforslag

Oppgave 4

To deloppgaver

Deloppgave 1 (10p)

I deloppgave 1 skal kandidaten vise at han/hun kan designe en enkel funksjon etter en gitt algoritme / formel. Sentralt her er at kand. Viser forståelse for argumentoverføring og returverdier. Et annet moment er å kunne håndtere overgang mellom int og float.

```
float bmi(float w, int h) // funksjonsdefinisjon
/*
    w:                vekt i kg.
    h:                høyde i hele cm.
    funksjonsverdi:  beregnet BMI
*/
{
    float value, hMeter;
    hMeter = (h/100.0);
    value = w / (hMeter * hMeter);
    return value;
}
```

Deloppgave 2 (10p)

I deloppgave 2 skal kandidaten vise hvordan den konstruerte funksjonen kan benyttes i et program. Kandidaten skal vise at han/hun kan skille mellom aktuell og formelle argumenter for en funksjon/funksjonskall.

```
#include <stdio.h>

float bmi(float w, int h); // funksjonsprototype

int main()
{
    int length;
    float weight;
    float BMI;

    printf("BMI program!\n");
    printf("Skriv inn vekt i kg (med desimaler): ");
    scanf("%f", &weight);
    printf("Skriv inn lengde i centimeter (uten desimaler): ");
    scanf("%d", &length);

    BMI = bmi(weight, length);

    printf("Din BMI er %.1f\n\n", BMI);
    return 0;
}
```

Oppgave 5

3 deloppgaver som er uavhengige av hverandre. Kandidaten skal kunne designe enkle programmer ut fra en gitt problemspesifikasjon. Som målsystem for deloppgavene benyttes **MBED LPC1768**.

Deloppgave 1 (10p)

Dette er en standardoppgave som kandidaten bør kjenne til fra labarbeidet.

Momenter:

- Sette opp en løkke (start- og stopp-betingelser)
- Deklarere BusOut-objekter
- Forstå at tallverdier som skrives til en BusOut objekt blir presentert på binærform på diodene.

```
//=====
#include "mbed.h"

BusOut bar(p21, p22, p23, p24, p25, p26, p27); // LSB: p21
DigitalOut blue1(LED1);

int main()
{
    blue1 = 0;
    for (int n = 1; n <= 100; n++) {
        bar = n;
        wait(0.1);
    }
    blue1 = 1;
    return 0;
}
//=====
```

Deloppgave 2 (10p)

Kandidaten bør kunne håndtere analog IO og kunne skille mellom avleste verdier og spenningsverdiene disse representerer. Det er viktig å vise at avlesning og beregning av spenningsverdi skjer innenfor løkka.

Kandidaten må også kunne generere pulser med ulik periodetid avhengig av de avleste måleverdiene.

```
//=====
#include "mbed.h"

AnalogIn ain(p16);
DigitalOut red(p18);

int main()
{
    float voltage;
    float pulseTime;

    while(1) {
        voltage = ain * 3.3;

        if (voltage > 3.0) {
            pulseTime = 0.1/2;
        } else if (voltage > 2.0) {
            pulseTime = 0.5/2;
        } else {
            pulseTime = 2.0/2;
        }

        red = 1;
        wait(pulseTime);
        red = 0;
        wait(pulseTime);
    }
}
//=====
```

Deloppgave 3 (10p)

Kandidaten må kunne kjenne til hvordan tid kan måles. Det enkleste vil være å benytte et Timer-objekt, men andre alternativer vil kunne være mulig å tenke seg.

Kandidaten viser bruk av venteløkker.

Kandidat bør vise at LED må tennes og slukkes for hver periode.

Kandidat må også kunne presentere informasjon til PC (terminalprogram) over USB.

```
//=====
#include "mbed.h"

DigitalOut blue(LED1);
DigitalIn D_1(p13), D_2(p14);

Serial pc(USBTX, USBRX);
Timer tim;

int main()
{
    int num = 1;
    pc.printf("MEASUREMENTS OF TDIFF IN PROGRESS:\n");
    while(1) {
        blue = 0;
        tim.reset();
        while(D_1 == 0) {
        }
        tim.start();
        blue = 1;
        while(D_2 == 0) {
        }
        pc.printf("Tdiff (%d) = %.2f sec.\n", num, tim.read());
        num++;
    }
}

//=====
```

2.1 En mikrokontroller er:

Velg alternativet som passer best!

- En miniatyrdatamaskin, som bare mangler CPU-en
- En databrikke som inneholder en CPU
- En miniatyrisert PLS (Programmerbar logisk styring)
- En databrikke som inneholder en komplett datamaskin med nødvendige enheter for å kunne kjøre programmer
- Kontrollenheten i en CPU
- En borddatamaskin fra 80-tallet

Maks poeng: 2.5

2.2 En datatype i C/C++ forteller noe om:

Velg alternativet som passer best!

- programmet er et C- eller et C++-program
- om en variabel kan endre verdi eller om den er konstant
- variabelen er lokal eller global
- hvor store program som kan lages
- hva slags informasjon som kan lagres i en variabel
- hvor mange ganger en løkke skal gjennomløpes

Maks poeng: 2.5

2.3 En lokal variabels navn gjelder (er synlig):

Velg alternativet som passer best!

- alltid innenfor filen den er definert
- i alle funksjoner
- kun innenfor funksjonen den er definert, hvis den ikke også er definert globalt
- kun innenfor funksjonen den er definert
- innenfor funksjonen den er definert og i den etterfølgende delen av programmet
- alltid innenfor main-funksjonen

Maks poeng: 2.5

2.4 I C er en tekststreng definert som:

Velg alternativet som passer best!

- en tabell (*array*) av typen *char* der antall tegn i strengen er lagret som et tall på posisjon 0
- en funksjon som behandler tekstinformasjon fra filer
- en variabel med navn som innledes med bokstavene *txt*
- en tabell (*array*) av typen *char* der et tegn med verdi 0 angir strengens avslutning
- en *struct* med datamedlemmer som hver representerer en bokstav
- en fil som inneholder tekst

Maks poeng: 2.5

2.5 En **if-else**-test benyttes i et C/C++-program fordi:

Velg alternativet som passer best!

- programmet skal vente til en bryter aktiveres
- programmet skal avsluttes umiddelbart
- programmet kan følge 2 av 3 alternative veier videre
- programmet skal utføre en instruksjonsblokk 8 ganger
- programmet kan velge å utføre én av 2 alternative oppgaver
- programmet kan velge å utføre én av 5 alternative oppgaver

Maks poeng: 2.5

2.6 Du ønsker å opprette en variabel som kan lagre verdien **-9999999** med 100% nøyaktighet. Hvilket av alternativene nedenfor kan du velge for å oppnå dette? Anta at det gjelder et MBED-program.

Velg ett alternativ

- unsigned int x;
- char x;
- int x;
- double x;
- float x;
- unsigned char x;

Maks poeng: 2.5

2.7 Gitt MBED-klassen AnalogIn.

Velg ett alternativ

- En analog spenning i området 0.0 V til 5.0 V kan omformes til et flyttall i området 0.0 til 1.0.
- En analog spenning i området 0.0 V til 3.3 V kan omformes til et heltall i området 0 til 33.
- En analog spenning i området 0.0 V til 3.3 V kan omformes til et flyttall i området 0.0 til 1.0.
- En analog spenning i området 0.0 V til 1.0 V kan omformes til et flyttall i området 0.0 til 3.3.
- En analog spenning i området 0.0 V til 3.3 V kan omformes til et flyttall i området 0.0 til 3.3.
- En analog spenning i området 0.0 V til 33.0 V kan omformes til et heltall i området 0 til 3.

Maks poeng: 2.5

2.8 Gitt MBED-klassen Timer:

Velg ett alternativ

- Denne klassen holder rede på dato og tid for mikrokontrolleren.
- Man kan benytte maks 1 Timer per program.
- Denne klassen måler medgått tid i sekunder, millisekunder eller mikrosekunder.
- En Timer må stoppes, før den kan avleses.
- Det er ikke mulig å stoppe en Timer når den først har startet.
- Timeren må alltid knyttes opp mot en digital inngang.

Maks poeng: 2.5

2.9 I en mikrokontroller benyttes følgende teknikk for å representere og regne med negative heltall:

Velg ett alternativ

- 2's komplement
- Fixed point
- Floating point
- 1's komplement
- Fortegn + tallverdi
- 10's komplement

Maks poeng: 2.5

2.10 Hvilke av alternativene er gyldige datatyper i C/C++?

Velg et eller flere alternativer

- int
- single
- negative
- double
- float
- real

Maks poeng: 2.5

2.11 Hvilke av alternativene angir løkke-typer i C/C++?

Velg et eller flere alternativer

- case
- union
- if
- for
- do-while
- continue

Maks poeng: 2.5

2.12 Hvilke(n) av følgende ord er navn på MBED-klasser og ikke standard i C++?

Velg et eller flere alternativer

- vector
- FILE
- Ticker
- BusOut
- stream
- Serial
- boolean
- string

Maks poeng: 2.5

2.13 I hvilke(n) av følgende uttrykk vil X få verdien 1?

Velg et eller flere alternativer

- $X = 9 \% 2;$
- $X = 9 \% 3;$
- $X = 3 - 1 * 1;$
- $X = 9 / 10;$
- $X = (3 != 0) + (1 > 2);$
- $X = 9 / 8;$

Maks poeng: 2.5

2.14 Hvilke(n) av følgende kan benyttes til å avslutte en løkke, men ikke et program?

Velg et eller flere alternativer

- break
- exit
- stop
- continue
- quit
- brexit

Maks poeng: 2.5

2.15 Hva er sant i forbindelse med bruk av funksjoner?

Velg et eller flere alternativer

- Programmene tar vesentlig større plass i minnet.
- Vanskelig å få oversikt i totalprogrammet.
- Enklere å teste ut deler av et program.
- Programmet blir raskere.
- Gjør det lettere å lage en god struktur i programmet.
- Vanskeligere å finne feil.

Maks poeng: 2.5

2.16 Hvordan kan verdier overføres fra "hovedprogram" til en funksjon?

Velg et eller flere alternativer

- Ved hjelp av konstanter
- Ved hjelp av datatyper
- Ved hjelp av globale variable
- Ved hjelp av tekststrenger
- Ved hjelp av funksjonsargumenter
- Ved hjelp av lokale variable

Maks poeng: 2.5

2.17 Hvilken datatype kan du velge for en variabel som skal inneholde verdier mellom **-500.00** og **+100000.00** med 2 desimalers nøyaktighet.

Velg null, ett eller flere alternativer

- float
- char
- short
- int
- double
- unsigned long

Maks poeng: 2.5

2.18 Hva er riktig i forhold til et objekt av klassen Ticker, etter at attach-funksjonen er utført?

Velg ett eller flere alternativer

- Lager en klokkepuls
- Avslutter programmet
- Måler tiden mellom to pulser
- Starter en funksjon om et gitt antall sekunder
- Lager en lyd hvert sekund
- Starter en funksjon hver gang det har gått et gitt antall sekunder

Maks poeng: 2.5

2.19 Hvis du ønsker å gjenta 5 kodelinjer 10 ganger: hvilke kontrollstrukturer i C/C++ er det da mulig å benytte?

Velg ett eller flere alternativer

- do-while løkke
- for-løkke
- if-else test
- while-løkke
- en funksjon
- switch-case

Maks poeng: 2.5

2.20 Hva er riktig påstand(er) i forhold til et C/C++ program?

Velg ett eller flere alternativer

- main-funksjonen kommer som oftest først i programmet
- Hvis to main-funksjoner benyttes, må den ene komme først i programmet og den andre sist.
- Programmet må inneholde nøyaktig én main-funksjon
- main-funksjonen kan komme først i programmet
- main-funksjonen må komme først i programmet
- Programmet må inneholde minst én main-funksjon

Maks poeng: 2.5

2.21 Hva er sant vedrørende en pekervariabel i C/C++?

Velg ett eller flere alternativer

- Kan multipliseres med en annen pekervariabel
- Kan inneholde adressen til en variabel
- Kan inneholde adressen til nest siste element i en tabell (array)
- En gitt pekevariable kan inneholde adressen til variable av ulike datatyper
- Kan kun inneholde adressen til heltallsvariable
- Kan inneholde adressen til en funksjon

Maks poeng: 2.5

2.22 Angivelse av konstanter.

Hvilke av eksemplene nedenfor angir gyldige konstanter i C/C++?

Velg ett eller flere alternativer

- 0x12G
- 430.56
- 095
- 0b11021
- 95
- 0xAA0

Maks poeng: 2.5

2.23 Studer følgende programlinjer:

```
int *p;  
int v = 9, u = -1;  
p = &u;  
v = *p * u;
```

Hva er riktig innhold i u og v etter at linjene er utført?

Velg ett alternativ

- $u = -1 / v = 1$
- $u = 9 / v = 1$
- $u = 9 / v = -9$
- $u = 1 / v = 1$
- $u = 0 / v = 0$
- $u = 0 / v = 9$

Maks poeng: 2.5

2.24 Anta at følgende deklarasjoner er gjort:

```
int a = 1, b = 2, c = 3;
```

Hvilke av følgende uttrykk er gyldige uttrykk i C/C++?

Velg ett eller flere alternativer

(1 && a)**

(b - 1 * c)

(a++ > 7)

(1 + 2.0)

(2 <<= b)

(1 >> 1) * (c << 5)

Maks poeng: 2.5

2.25 Hva betyr det at mikrokontrolleren "LPC 1768 ARM Cortex M3" som vi finner på MBED-modulen, er en 32-bits mikrokontroller?

Velg ett alternativ

Størrelsen på minnet er 32 Mbytes

Klokkehastigheten er 32 MHz

Størrelsen på interne arbeidsregistre er 32 bit

Det er 32 arbeidsregistre

Antall funksjoner som et program kan bestå av er 32

Antall maskininstruksjoner er 32

Maks poeng: 2.5

- 3.1 Nedenfor ser du en del av et program som skal kjøres på en PC.
Hvilke(n) av programlinjene inneholder **syntaksfeil**? Dvs. feil som fører til kompileringssfeil.

Anta at det ikke er noen problemer i forhold til innplasseringen i programmet forøvrig, og at aktuelle headerfiler er inkludert.

Velg ett eller flere alternativer

- int n;
- int x = 1 mangler semikolon
- for (n = 0; n < 4, n++) { komma skal erstattes med semikolon mellom 2. og 3. ledd.
- scanf("%d", x); må ha & foran variabelnavn
- if (x == 0)
- Printf("Svaret er 0\n"); printf med liten p
- else
- printf("Svaret er noe annet\n");
- else en else som henger for seg selv
- printf("Svaret er helt noe annet\n");
- }

Maks poeng: 5

- 3.2 Studér denne programsnutten:

```
float v = 37.3, x = 0.0;
while (x < v)
{
    x = x + 3.5;
}
```

Skriver et inn et tall som angir verdien til **x** etter at løkka er ferdig: .

Studér så denne programsnutten:

```
int a = 5, b = 13, t;
if (a > 6) {
    b = 7;
}
else {
    t = a;
    a = b;
    b = t;
}
```

Skriver et inn et tall som angir verdien til **b** etter at programlinjene er utført: .

Maks poeng: 5

3.3 Studér denne kodesnutten:

```
int n, m, k = 1 ;
for (n = 1; n <= 8; n++) {
    k = k * 2;
}
m = n;
while (m > 0) {
    if (m < 5)
        break;
    m = m - 1;
}
```

Hvor mange ganger utføres for-løkka: .

Hva er verdien til **k** etter at programlinjene er utført: .

Hva er verdien til **m** etter at while-løkka er avsluttet: .

Maks poeng: 5

3.4 Studér følgende kodesnutt:

```
int x = 0, n = 10;
int arr[ 10 ] = {1, 0, -1, 2, 0, -2, 3, 0, -3, 25};

while( n ) {
    x += arr[n-1];
    n = n - 1;
}
```

Skriv inn verdien til **x** etter at koden er utført: .

Skriv inn verdien til **n** etter at koden er utført: .

Maks poeng: 5